

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS FÍSICAS
Departamento de Informática y Automática



TESIS DOCTORAL

**Diseño automático de sistemas digitales : métodos
heurísticos para la colocación óptima de elementos**

MEMORIA PARA OPTAR AL GRADO DE DOCTOR
PRESENTADA POR

José Luis San Emeterio Prieto

DIRECTOR:

Ricardo García Rosa

Madrid, 2015

José Luis San Emeterio Prieto



5309856381

**DISEÑO AUTOMÁTICO DE SISTEMAS DIGITALES: MÉTODOS HEURÍSTICOS
PARA LA COLOCACIÓN ÓPTIMA DE ELEMENTOS**

**Departamento de Informática y Automática
Facultad de Ciencias Físicas
Universidad Complutense de Madrid
1981**



BIBLIOTECA

TP
1981
167

© José Luis San Emeterio Prieto
Edita e imprime la Editorial de la Universidad
Complutense de Madrid. Servicio de Reprografía
Noviciado, 3 Madrid-8
Madrid, 1981
Xerox 9200 XB 480
Depósito Legal: M-23081-1981

DISEÑO AUTOMÁTICO DE SISTEMAS DIGITALES: MÉTODOS HEURÍSTICOS
PARA LA COLOCACIÓN ÓPTIMA DE ELEMENTOS

Memoria presentada por D. José
Luis San Emeterio Prieto, para
optar al grado de Doctor en --
Ciencias Físicas.

Universidad Complutense de Madrid, Facultad de Ciencias Físicas.

A Pilar

Quiero expresar mi agradecimiento a todos los compañeros del Instituto de Automática Industrial que han colaborado y me han animado en la realización de este trabajo. - De una manera especial a todos los componentes de la sección de Informática, cuya ayuda facilitó su terminación, a Conchita y Charo por su esfuerzo en el mecanografiado.

Igualmente mi gratitud a Ricardo García Rosa que - lo ha dirigido y al profesor D. Antonio Vaquero Sánchez por su asesoramiento y ayuda en todos los ordenes.

Todos ellos ha hecho posible la presentación de - esta memoria.

I N D I C E

	<u>Página</u>
- Propósito y desarrollo	1
- Introducción	9
- Capítulo I : Planteamiento general	11
- Capítulo II : Colocación inicial	23
2.1. Colocación inicial constructiva	23
2.2. Colocación inicial aleatoria	27
2.3. Retículas	27
2.4. Descripción de los ejemplos estudiados.	29
2.5. Resultados obtenidos	32
- Capítulo III : El problema de Asignación Cuadrática.	35
3.1. Definición formal del problema	35
3.2. Antecedentes: Métodos conocidos más importantes	37
3.2.1. Métodos semienumerativos	38
3.2.2. Métodos iterativos	40
3.3. Intercambio por parejas	47
3.3.1. Técnicas de relajación en el intercambio por parejas	49
3.4. Una nueva estrategia	52

3.5. Cálculo incremental para la función de evaluación	56
3.6. Algoritmos contruidos para realizar un estudio comparativo	60
3.7. Fluctuación de los resultados con el factor de vecindad	62
3.8. Resultados obtenidos: Análisis y con-- clusiones	65
- Capítulo IV : El Problema de colocación ...	73
4.1. Métodos iterativos para tratar directa <u>m</u> ente el problema de colocación	74
4.1.1. Algoritmo utilizado	76
4.2. Métodos iterativos para tratar el pro- blema de colocación por transformación en asignación cuadrática	79
4.2.1. Transformación clásica	81
4.2.2. Una transformación nueva	81
4.3. Ejemplos estudiados y resultados obte- nidos	85
- Capítulo V : El problema de colocación en el diseño automático de tarje <u>s</u> tas de circuitos impresos ...	92
5.1. Lenguaje orientado al problema	94
5.2. Disponibilidad de pistas: Nueva defini <u>c</u> ción de distancia	98

5.3. Comparación de las colocaciones obtenidas por distintos métodos	102
- Principales aportaciones y conclusiones	122
- Apéndice I	126
- Apéndice II	137
- Bibliografía	142

PROPOSITO Y DESARROLLO

El diseño automático se basa en la utilización de los ordenadores digitales como herramienta para la automatización - de un proceso de diseño. Su objetivo fundamental es la reducción del coste y del tiempo necesarios desde el comienzo del diseño - hasta la terminación de la fabricación.

Desde sus orígenes, en la mitad de la década 1950-60, hasta la fecha, el diseño automático de sistemas digitales se ha convertido en un área de estudio extensa e importante. La complejidad alcanzada por los sistemas digitales ha llegado a un nivel en el que es virtualmente imposible, en ciertos casos, obtener - un diseño correcto sin hacer uso de las ayudas de diseño automático. Por otra parte, estas ayudas se extienden a las diversas - fases del proceso: Síntesis, simulación, diseño físico y verificación (1).

Terminado un diseño lógico, su diseño físico requiere la participación y asignación de la lógica a unidades físicas, - la colocación de estas unidades físicas en unidades funcionales mayores (las cuales a su vez pueden ser parte de otras unidades mayores, que también deberán ser colocadas adecuadamente) y la realización de las conexiones de acuerdo con el diseño lógico. El planteamiento a este nivel de abstracción nos permite expresar, con independencia del desarrollo de la tecnología, los tres problemas fundamentales que se presentan en el diseño físico: Partición-asignación, colocación y trazado de conexiones.

Aunque estos tres problemas están estrechamente relacionados, han sido siempre tratados por separado, debido a la gran complejidad inherente a cada uno de ellos.

El estado actual de la tecnología con la integración a gran escala, tendiendo continuamente hacia niveles más altos de integración, y las tarjetas de circuitos impresos multicapa, permitiendo una compactación creciente de las pastillas, ha hecho aumentar la importancia del problema de colocación exigiendo nuevos algoritmos que permitan tratar casos con un número de elementos cada vez más elevado.

La colocación en el diseño automático de sistemas digitales se plantea a tres niveles: Colocación de puertas lógicas en pastillas de circuitos integrados, colocación de pastillas en tarjetas de circuitos impresos y colocación de tarjetas en armarios.

El problema de colocación consiste esencialmente en asignar de una manera óptima un conjunto de elementos, interrelacionados N a N , siendo N variable, a un conjunto de posibles posiciones. El caso particular en el que N es constante con valor 2, recibe el nombre de asignación cuadrática.

Es interesante señalar que este problema se presenta también en otras áreas distintas al diseño automático de sistemas digitales, como por ejemplo la planificación de un área urbana, el diseño de una planta industrial o el almacenamiento de

3.-

registros en un medio de acceso no directo. En cada caso la optimización se realizará en base a una función de evaluación adecuada.

Desde el punto de vista de su resolución, el problema de colocación presenta esencialmente dos aspectos: 1º) Establecimiento de un algoritmo de optimización y 2º) Determinación de la función de evaluación. Nos podemos hacer una idea del grado de dificultad que plantea el primer aspecto teniendo en cuenta, que no se conoce solución al problema de asignación cuadrática, cuando el número de elementos que intervienen supera el rango de 15 ó 20 elementos, y que la solución al problema de asignación cuadrática es bastante más sencilla que la del problema de colocación. La dificultad en el segundo aspecto reside en la transformación en una función matemática evaluable de un objetivo global, como por ejemplo puede ser el facilitar el trazado de conexiones.

Dentro de todo el contexto anterior se ha planteado el trabajo de investigación, que ha dado lugar a la presente memoria.

El punto de partida, ha sido un trabajo anterior (2) que nos ha permitido disponer de un algoritmo para trazado de conexiones, que permite unir un punto a un conjunto de puntos previamente conectados, y de un lenguaje para la descripción de tarjetas de circuitos impresos.

Las tarjetas, procedentes de distintos usuarios, que han sido diseñadas utilizando el sistema, nos ha permitido -

disponer de problemas reales de colocación.

El algoritmo de trazado de conexiones es la herramienta adecuada para medir la efectividad de una colocación en el diseño automático de tarjetas de circuitos impresos.

Los objetivos fundamentales que nos hemos propuesto, son los siguientes:

- 1º .- Establecimiento de una heurística racional, válida para resolver el problema de colocación con un número elevado de elementos, en un tiempo razonable, haciendo un análisis del comportamiento del algoritmo.
- 2º.- Determinación de la función de evaluación adecuada para el diseño automático de tarjetas de circuitos impresos, basada en los siguientes objetivos:
 - a) Facilitar el trazado de las conexiones.
 - b) Minimizar el número de taladros necesarios.
 - c) Minimizar la longitud de conexionado.
- 3º .- Integrar los programas de colocación en un sistema para diseño automático de tarjetas de circuitos impresos, que incluya colocación y trazado de conexiones.

Para realizar este trabajo se ha dispuesto de un miniordenador como herramienta fundamental, por lo que la optimización de la memoria requerida ha sido siempre un factor esencial.

La memoria que se presenta consta de una breve introducción, cinco capítulos y dos apéndices.

En el primer capítulo, se presenta la definición formal y genérica del problema de colocación y se concretan algunos aspectos de esta definición adaptandolos al caso del diseño automático de sistemas digitales. Se establece una función de evaluación dependiente del tipo de árbol que une un conjunto de puntos y se analizan los principales tipos de árboles usados en diseño automático. También se presenta el problema de asignación cuadrática, como caso particular y como problema asociado al de colocación. Se describen los métodos constructivos e iterativos como métodos generales de resolución de los problemas de asignación cuadrática y colocación. Por último, se razona el enfoque dado al problema, descomponiéndolo en dos partes: a) encontrar un buen algoritmo para resolver el problema de asignación cuadrática y b) encontrar una buena transformación del problema de colocación en asignación cuadrática.

En el segundo capítulo, se describen los dos métodos de colocación inicial, que se van a utilizar a lo largo de todo el trabajo. El primero está basado en la utilización de un generador de números aleatorios. El segundo es un método constructivo.

También se describen las retículas, que definen el conjunto de posiciones disponibles para los distintos ejemplos que se van a estudiar en los capítulos III y IV.

Por último, se presentan los ejemplos que se van a usar en el estudio del problema de asignación cuadrática, procedentes de los problemas de colocación descritos en el apen-dice II, y los resultados obtenidos en cada caso.

El tercer capítulo comienza con una definición for-mal del problema de asignación cuadrática, analiza sus antecedentes y los métodos generales más importantes que se conocen para su tratamiento.

Se razona la elección de un método basado en aplicar técnicas de relajación al intercambio por parejas y se describen los factores que influyen en el comportamiento de este método.

Se presenta una nueva estrategia para la selección - del elemento secundario que refuerza a la heurística básica de relajación. Se presenta también un método para ajuste automatico del factor de vecindad, y se desarrolla un método para el - cálculo incremental de la función de evaluación.

Por último se realiza un estudio comparativo entre - los nuevos algoritmos y otros ya conocidos, y se analiza la influencia del factor de vecindad.

En el cuarto capítulo se estudia el problema de colocación, utilizando una función de evaluación basada en el árbol de expansión mínima. Se utiliza un algoritmo basado en el inter

cambio exhaustivo por parejas, para el tratamiento directo del problema.

Se describe el tratamiento del problema de colocación por transformación clásica en asignación cuadrática y se presenta una nueva función de transformación.

Por último se realiza un estudio comparativo, del comportamiento de los tres métodos, con los ejemplos descritos en el apéndice II, comprobándose el buen comportamiento de la transformación propuesta.

En el quinto capítulo se estudia el problema de colocación en el diseño automático de tarjetas de circuitos impresos, se analiza la transformación del problema real de colocación de componentes al problema abstracto de colocación de elementos en posiciones. Se presenta una nueva definición de distancia capaz de incorporar a la función de evaluación basada en el árbol de expansión mínima, una información sobre la disponibilidad de pistas.

Por último se estudian los ejemplos, presentados en el apéndice I, midiendo los resultados en base al número de puntos no conectados, número de taladros necesarios y longitud total de conexión, y se presenta el diseño obtenido por el sistema completo con colocación y trazado de conexiones automáticas.

El apéndice I, contiene la descripción en el lenguaje

8.-

definido en (2), de tres de las cuatro tarjetas utilizadas. Se explican sus características y el motivo de su elección. Para la cuarta tarjeta, procedente de la unidad de control del ordenador ILLIAC IV, se especifica su origen y características.

El apendice II, contiene la descripción de los problemas de colocación que se derivan de las tarjetas presentadas en el apendice I.

INTRODUCCION

Existe un algoritmo sencillo para resolver tanto el problema de colocación como el de asignación cuadrática, consistente en ir generando ordenadamente todas las posibles colocaciones o asignaciones y calculando para cada una de ellas el valor de la función de evaluación. Conservando el mínimo de estos valores y su colocación asociada obtenemos la solución exacta del problema. Se trata evidentemente de un algoritmo enumerativo.

Este método, aunque teóricamente, posible, no es factible en la práctica, a poco elevado que sea el número de elementos, puesto que si intervienen n elementos y n posiciones, necesitamos calcular $n!$ veces la función de evaluación.

Existen métodos semienumerativos, capaces de proporcionar la solución exacta al problema de asignación cuadrática para casos con un número de elementos no superior a 20.

Para los problemas reales que actualmente se presentan, con un número de elementos cada vez más elevado, es imprescindible recurrir a métodos heurísticos, que proporcionan una solución aproximada, localmente óptima en el sentido de que es el valor mínimo de la función de evaluación, que se puede obtener con la heurística adoptada y los parámetros que la concretan.

La dificultad fundamental para evaluar el comportamiento de estos métodos heurísticos reside en que nunca sabemos la proximidad de la solución que proporcionan a la solución exacta del problema.

Cuando se pretende desarrollar nuevos métodos heurísticos para estos problemas, es imprescindible por tanto, comparar su comportamiento con el de otros métodos ya conocidos, siendo necesaria la programación de estos últimos. Al mismo tiempo, es necesario utilizar casos concretos con un número de elementos que cubra el rango que queremos estudiar.

Con este fin, a lo largo de este trabajo, se han programado algoritmos ya conocidos para realizar un estudio comparativo en cada etapa y se han utilizado varios ejemplos procedentes de problemas reales de diseño automático.

C A P I T U L O I

PLANTEAMIENTO GENERAL

Podemos definir, desde un punto de vista formal, el - problema de colocación de la siguiente manera:

Sea $V = \{v_1, v_2, \dots, v_n\}$ un conjunto de n elementos y $P = \{p_1, p_2, \dots, p_r\}$ un conjunto de r puntos pertenecientes a un espacio métrico en el que está definida la distancia d , y tales que $r \gg n$.

Definimos una conexión c como un subconjunto de V , -- con un número de elementos mayor o igual a dos. Sea $C = \{c_1, c_2, \dots, c_m\}$ un conjunto de m conexiones definidas sobre V , tal -- que los subconjuntos c_i de V son no disjuntos. Es decir un elemento puede pertenecer a varias conexiones.

Dado un subconjunto K de puntos de P , existen numerosos árboles correspondientes a distintos tipos de árbol, que unen dichos puntos. Para cada uno de estos árboles la longitud de sus ramas se puede calcular a partir de la distancia d .

Denominamos longitud de un árbol a la suma de las longitudes de sus ramas.

Dado un tipo de árbol T , denominaremos $|T_K|$ a la longitud de aquel, o aquellos árboles que correspondiendo al tipo

T, unen los puntos de K con una longitud mínima.

Sea s una aplicación inyectiva del conjunto V en el conjunto P ; s define una colocación de los n elementos de V en los r puntos (posiciones) de P . Sea S el conjunto de todas las posibles aplicaciones inyectivas de P en V . El conjunto S tiene $r!/(r-n)!$ elementos.

A partir de una aplicación s podemos definir una correspondencia del conjunto C en el conjunto P , de la siguiente manera:

$$z(c_i) = \{s(v_j) \mid v_j \in c_i\}$$

Para cada elemento $s \in S$ definimos una función de evaluación $F(s)$ de la siguiente forma:

$$F(s) = \sum_{c_i \in C} |T_z(c_i)|$$

El problema de colocación consiste en encontrar el elemento $s^* \in S$ tal que

$$F(s^*) = \min \{F(s) \mid s \in S\}$$

Tal como se ha definido, la función de evaluación depende esencialmente de la definición de distancia entre dos puntos y del tipo de árbol que une un conjunto de puntos. Si se desea una definición más general del problema se debería de

jar la función de evaluación $F(s)$ sin definir, abierta a la influencia de otros factores.

Sin embargo consideramos que para el diseño automático de sistemas digitales, que es el campo en el que el problema de colocación ha despertado más interés y ha recibido más esfuerzo, la definición que hemos dado es suficientemente amplia. En efecto, el objetivo fundamental de la colocación en el diseño automático es permitir que se puedan realizar todas las conexiones especificadas minimizando el número de cruces entre conexiones y la longitud total de conexionado. Tradicionalmente se ha considerado que la minimización de la longitud total de conexionado es la meta para resolver globalmente los diversos objetivos, siendo la medida usual para evaluar los algoritmos de colocación.

Actualmente se están realizando esfuerzos para incluir otros factores en la función de evaluación, tales como la congestión de pistas en una zona determinada (3). Estos métodos se basan en obtener varias colocaciones finales, basadas en una función de evaluación tal como la que hemos definido aquí y compararlas posteriormente con una función de evaluación más sofisticada, que incluye normalmente un algoritmo rápido y aproximado de conexionado. En base a esta última evaluación se seleccionaría una de las varias colocaciones generadas. Está claro sin embargo que la base fundamental de estos algoritmos radica en una función de evaluación del tipo que aquí hemos definido.

Por otra parte existe en el diseño de sistemas digitales un conjunto de factores que deben ser tomados en cuenta, por ejemplo: la eliminación de interferencias sin hacer un uso indebido de técnicas costosas de apantallamiento o la disipación térmica no permitiendo concentraciones de fuentes de calor. Desde un punto de vista práctico no es posible incluir estos factores junto a los anteriores en una función de evaluación. No obstante, esto no produce graves problemas, porque generalmente puede ser tratados introduciendo ligaduras o restricciones en los algoritmos de colocación y conexionado.

Por último, existe la posibilidad de introducir -- otros factores, como la disponibilidad de pistas por ejemplo, en la función de evaluación a través de una definición adecuada de distancia.

En resumen, consideramos que la definición dada es lo suficientemente amplia y adecuada al problema de colocación en el diseño automático de sistemas digitales, siendo incluso necesario concretar más algunos aspectos:

- A). En el caso más frecuente el espacio métrico coincide con R^2 . A partir de ahora en este trabajo siempre utilizaremos implícitamente R^2 .
- B). La definición de distancia más adecuada al diseño automático es $d_{ij} = |x_i - x_j| + |y_i - y_j|$. Mientras no se indique lo contrario será esta la distancia que utilizaremos.

C). El tipo de árbol que une un conjunto K de puntos de P, depende fundamentalmente de la tecnología utilizada y debe estar estrechamente relacionado con el algoritmo de conexión que se vaya a utilizar después.

En el diseño automático de sistemas digitales se usan básicamente tres tipos de árboles:

1.- Árboles de expansión mínima fig. (1.a).

Son árboles cuyos nodos son los K puntos y cuya longitud es mínima.

2.- Árboles Steiner fig. (1.b).

Son árboles cuyos nodos incluyen los K puntos, pero pueden incluir otros puntos auxiliares como nodos y cuya longitud es mínima.

3.- Árboles especiales.

Son árboles que satisfacen ciertas exigencias en el grado de sus nodos u otras limitaciones topológicas.

3.a. Cadena mínima fig. (1.c).

Es un árbol que tiene como máximo dos ramas incidentes a cualquier nodo, y cuya longitud es mínima.

3.b. La figura 1.d representa un árbol tal que todos los nodos, excepto la raíz, son nodos terminales, es decir - tienen grado cero.

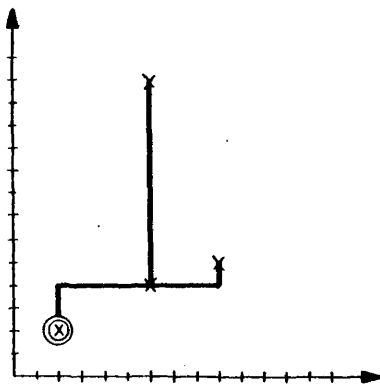
En la figura 1, se representan los diversos tipos de árboles que unen cuatro nodos señalados con un aspa. La raíz - (nodo fuente) está encerrada en cada caso con un doble círculo. Cuando se introducen nodos auxiliares se les señala con un solo círculo.

El desarrollo actual de la tecnología ha hecho caer en desuso la necesidad de la unión entre puntos siguiendo árboles especiales. En la actualidad prácticamente solo se usan -- los árboles de expansión mínima y los árboles Steiner.

El árbol que más se aproxima al trazado que realiza un diseñador manual y al problema real de conexión es el árbol Steiner, sin embargo es el más difícil de encontrar. De hecho el problema de determinar el árbol Steiner que une un conjunto n de puntos está sin resolver para un valor poco elevado de n y es muy costoso incluso cuando n es pequeño.

El árbol de expansión mínima, aunque también costoso de evaluar, lo es mucho menos que el correspondiente árbol -- Steiner (4) y por ello ha sido tradicionalmente utilizado como el tipo de árbol adecuado para definir la función de evaluación. Los pocos estudios comparativos de algoritmos de colocación que se conocen, el más importante de los cuales es el debido a M. Hanan (5), se basan en este tipo de árbol.

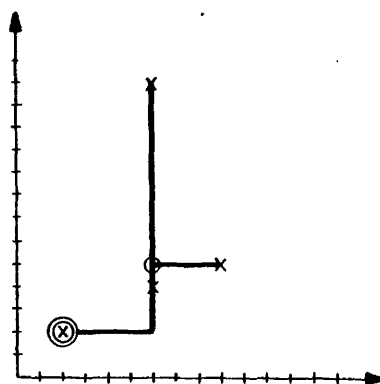
La diferencia fundamental entre el árbol Steiner y el árbol de expansión mínima radica en que para el primer caso



Arbol de expansión mínima

longitud = 19

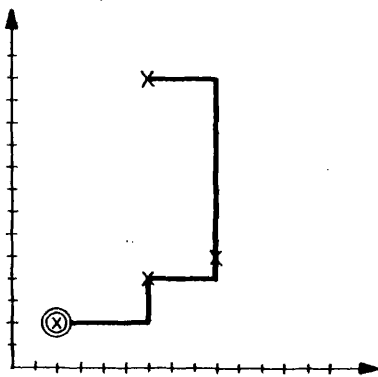
fig. 1.a



Arbol Steiner

longitud = 18

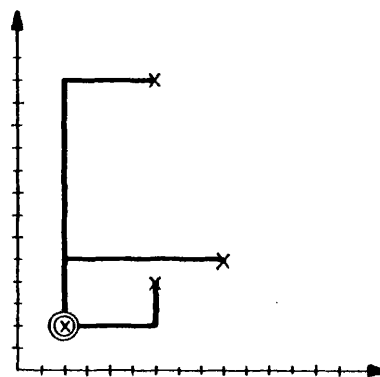
fig. 1.b



Cadena mínima

longitud = 21

fig. 1.c



Arbol de raíz a terminales

longitud = 31

fig. 1.d

existe un número infinito de posibles soluciones, mientras -- que para el segundo este número (aunque elevado) es finito.

Calcular el árbol de expansión mínima es un problema combinatorio.

Un caso particular, muy importante, del problema de colocación es el problema de asignación cuadrática. El -- problema de colocación coincide con el problema de asignación cuadrática cuando todas las conexiones de C se componen exactamente de dos elementos.

Este problema aunque sea un caso particular del problema de colocación y esté intimamente ligado a él, tiene una -- entidad propia, por lo que presentaremos su definición formal en el capítulo de asignación cuadrática.

A todo problema de colocación se le pueden asociar varios problemas de asignación cuadrática dependiendo de la -- función de transformación que se adopte. La más sencilla y la más utilizada consiste en hacer el número de interconexiones que existen entre dos elementos v_i y v_j , c_{ij} , igual al número de conexiones a las que pertenecen ambos elementos . Es decir:

$$c_{ij} = \sum_{c_1 \in C} w_1 \quad \text{siendo} \quad \begin{cases} w_1 = 1 \text{ cuando } v_i, v_j \in c_1 \\ w_1 = 0 \text{ en caso contrario.} \end{cases}$$

El tratamiento del problema de asignación cuadrática es más sencillo que el del problema de colocación. La razón fundamental reside en que la función de evaluación es bastante más fácil de calcular.

Por ello, es muy frecuente transformar el problema de colocación en un problema de asignación cuadrática asociado a él. La mayoría de los primeros trabajos se basaban en una transformación del tipo que hemos descrito. Actualmente se está dedicando bastante esfuerzo a buscar funciones de transformación más adecuadas.

Podemos clasificar los métodos para el tratamiento tanto del problema de colocación como del de asignación cuadrática en dos grandes grupos:

- Métodos constructivos.
- Métodos iterativos.

Los métodos constructivos consisten en realizar secuencialmente las siguientes operaciones:

- a) Seleccionar un elemento no colocado.
- b) Colocar el elemento seleccionado en una posición no ocupada.

Una vez que se ha colocado un elemento, quedará definitivamente asignado a su posición. Los métodos constructivos

no producen una colocación de todos los elementos hasta que el algoritmo correspondiente termina.

Los distintos métodos constructivos se diferencian en el criterio para seleccionar el elemento que se va a colocar de entre todos los no colocados y en el criterio para determinar en qué posición libre se va a colocar este elemento - (6).

La gran ventaja de los métodos constructivos reside en que el tiempo de cálculo que necesitan es pequeño, siendo la aproximación que proporcionan suficiente en algunas aplicaciones.

El principal inconveniente de los métodos constructivos radica en que la aproximación que proporcionan no es suficiente para muchas otras aplicaciones y la solución que proporcionan no puede ser modificada por un método constructivo para mejorarla. Por ello frecuentemente se usan los métodos constructivos con el fin de obtener una colocación inicial que será tratada posteriormente por un método iterativo.

Los métodos iterativos parten de una colocación inicial definida. Generalmente requieren un tiempo de cálculo más elevado que los métodos constructivos pero producen mejores resultados. En los métodos iterativos se selecciona un subconjunto de elementos, a los que se les va a permitir cambiar de posición, junto a sus posiciones asociadas. Estos elementos se colo

can de nuevo siguiendo unas normas tendentes a minimizar la función de evaluación. En cada paso de iteración o bien se obtiene una nueva colocación mejor que la anterior o bien se conserva la antigua. Los métodos iterativos van asociados a unas reglas de terminación del algoritmo.

La necesidad de los métodos iterativos para obtener una buena solución, las funciones de evaluación ligadas al problema de colocación, que requieren un elevado tiempo de cálculo, y la tendencia actual hacia problemas de colocación con un número elevado de elementos, determinan que la aproximación más razonable al problema sea transformarlo en un problema de asignación cuadrática asociado y resolver este último.

Ahora bien, una solución localmente óptima al problema de asignación cuadrática no es necesariamente buena al problema de colocación asociado.

En nuestra opinión, el que un buen algoritmo de asignación cuadrática resuelva eficazmente un problema de colocación depende esencialmente de la función de transformación que se adopte.

Es esta función de transformación la que debe tratar de que quede reflejado, lo más fielmente posible, en el problema de asignación cuadrática, el tipo de árbol que define la función de evaluación del problema de colocación. Por

ejemplo, consideramos que no es adecuada la función de transformación que hemos presentado en este capítulo y que por otra parte ha sido ampliamente utilizada.

Por todo ello, el enfoque que hemos dado al problema, ha consistido en descomponerlo en dos partes:

- a) Encontrar un buen algoritmo para resolver el problema de asignación cuadrática.
- b) Encontrar una buena función de transformación del problema de colocación en asignación cuadrática.

C A P I T U L O I I

COLOCACION INICIAL

Los métodos iterativos de colocación, que son imprescindibles para la obtención de un resultado suficientemente bueno, exigen como punto de partida una colocación inicial.

En este trabajo hemos utilizado dos métodos de colocación inicial basados en la transformación del problema de colocación en asignación cuadrática: un método constructivo y un método aleatorio.

2.1.- Colocación Inicial Constructiva

Como ya hemos dicho en el capítulo I, es muy frecuente la utilización de métodos constructivos para obtener una colocación inicial. De todas formas, siempre hemos de tener en cuenta que, una colocación obtenida de esta manera, puede ser por sí misma una solución suficientemente buena para algunas aplicaciones.

Los métodos constructivos se diferencian unos de otros en el criterio de selección y en el criterio de colocación.

El criterio de selección, que determina en cada paso - cual de los elementos no colocados va ser colocado, debe estar -

basado en alguna medida de las conexiones que existen entre los elementos no colocados y los colocados.

Existen dos criterios de selección, debidos a J. M. - Kurtzberg (7), que podemos resumir de la siguiente manera.

Sea A el conjunto de elementos ya colocados y B el conjunto de elementos no colocados. Definimos:

$$f_1(v_i) = \max_{v_j \in A} c_{ij}; \quad \forall v_i \in B$$

$$f_2(v_i) = \sum_{v_j \in A} c_{ij}; \quad \forall v_i \in B$$

El primer criterio de selección consiste en elegir el elemento de B que maximice f_1 . El segundo criterio selecciona - el elemento de B que maximiza f_2 .

El primer criterio busca el elemento más conectado a un elemento ya colocado. El segundo criterio busca el elemento más conectado al conjunto de los elementos ya colocados.

El criterio de colocación debe tratar de encontrar, para el elemento seleccionado, la posición que minimice la función

de evaluación, calculada sobre los elementos ya colocados, con un tiempo de cálculo razonable.

Dos criterios de colocación están ligados a los dos - criterios de selección expuestos. Al criterio de selección basado en f_1 le corresponde un criterio de colocación, que trata de llevar el elemento seleccionado, a la posición más cercana al - elemento al que está más conectado.

El criterio de colocación ligado a f_2 , trata de asignar el elemento seleccionado, a la posición más cercana al "centro de gravedad" de los elementos colocados, a los que está conectado.

A partir de los criterios expuestos, se pueden construir métodos de colocación inicial ligeramente diferentes, según el tratamiento que se dé a los casos, en los que la función de selección toma el mismo valor para varios elementos de B o - en los que el criterio de colocación determina varias posiciones como igualmente válidas para un elemento.

Existen y se pueden definir otros criterios, más o menos sofisticados, para la selección y la colocación (8). Sin embargo los que hemos presentado son los más utilizados y los - que conducen a mejores resultados como colocación inicial (9).

Hemos construido un método de colocación inicial - constructiva, utilizando el criterio de selección basado --

en f_1 y el criterio de colocación asociado, que podemos expresar de la siguiente manera:

- P1.- Inicialización. Inicializar A y B. Si existe algún elemento asignado a priori, de forma definitiva, a alguna posición, pasar a P2. En caso contrario seleccionar los dos elementos más conectados entre sí y colocarlos en las posiciones más próximas al centro del conjunto de puntos P. Actualizar A y B.
- P2.- Selección. Calcular la función f_1 , para cada elemento del conjunto B. Seleccionar el elemento que maximice f_1 . En caso de empate seleccionar el primer elemento que haya hecho máxima f_1 .
- P3.- Colocación. Situar el elemento seleccionado en P2, en la posición no ocupada, más próxima al elemento ya colocado, al cual está más conectado. En el caso de que existan varias posiciones, elegir la primera que se haya encontrado.
- P4.- Test de finalización. Actualizar los conjuntos A y B. Si el conjunto B está vacío, el algoritmo termina. En caso contrario pasar a P2.

2.2.- Colocación Inicial Aleatoria

Hemos construido otro método de colocación inicial utilizando un generador de números aleatorios. Podemos resumirlo en los siguientes pasos:

P1 .- Generar aleatoriamente N colocaciones

P2 .- Para cada colocación, calcular la función de evaluación $F(s)$.

P3 .- Tomamos como colocación inicial aquella cuya función de evaluación sea mínima. Terminación.

En este trabajo se ha utilizado $N = 10$

2.3.- Retículas

Para el estudio de los problemas de colocación y de asignación cuadrática, a nivel abstracto, el conjunto de puntos P que vamos a utilizar está constituido por los vértices de una retícula cuadrangular, de lado unidad, numerados de izquierda a derecha y de arriba a abajo. Describimos una retícula por su número de columnas NC y su número de filas NF.

Esta estructura es válida para el estudio de los casos reales en diseño automático, sin más que transformarla en retícula rectangular de lados DX y DY.

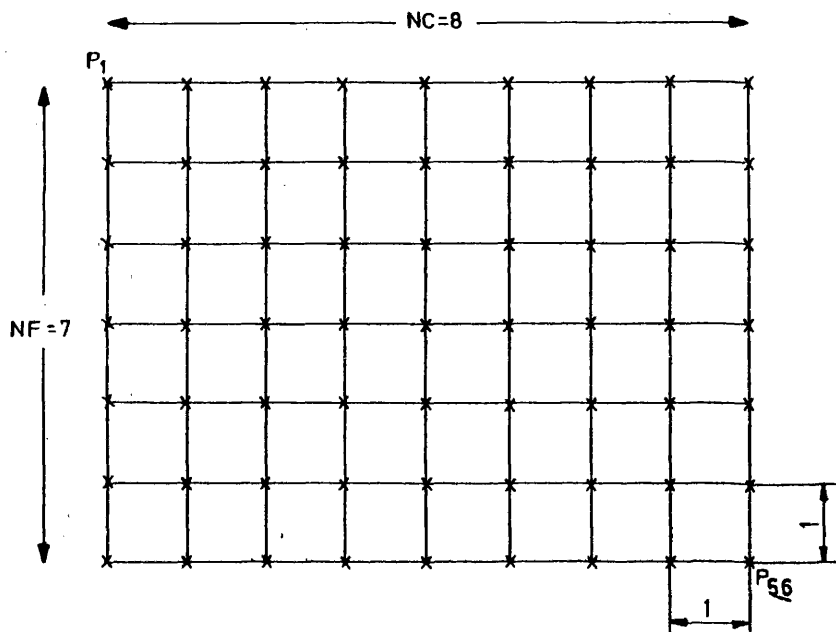


Fig.-2

La fig. 2 representa la retícula utilizada para el estudio del caso procedente de la tarjeta IKER3.

2.4.- Descripción de los ejemplos estudiados

Los ejemplos que vamos a estudiar proceden de los problemas reales de diseño de circuitos de tarjetas impresas, descritos en el Apéndice I. Cada tarjeta da origen a un caso de colocación. Para el estudio del problema de asignación cuadrática como tal, transformamos cada caso de colocación en uno de asignación cuadrática, utilizando la función de transformación presentada en el capítulo I.

- Ejemplo 1 : UCA

Número de elementos a colocar = 20

Número de elementos asignados a priori (fijos) = 5

Retícula : NC = 5 ; NF = 4

Elementos fijos:

Elemento	Posición
16	18
17	1
18	6
19	11
20	16

- Ejemplo 2 : STEIMBERG

Este caso es el único ejemplo disponible en la literatura científica, para estudios comparativos, hasta la publicación por J. Stevens (10) de la descripción de cinco tarjetas

de la unidad de control del ordenador ILLIAC IV.

Este ejemplo publicado por L. Steimberg (11), procede de un ordenador UNIVAC. Steimberg publicó el problema de colocación ya transformado en asignación cuadrática.

Queremos hacer notar, que en nuestra opinión existe un error de mecanografía, en la matriz de interconexiones de la página 43 de dicha publicación. El número de elementos a los que está conectado el elemento E29 es 8 en lugar de 7 y el número total de conexiones a las que pertenece dicho elemento es 79 en lugar de 69. Esto produce que el número total de interconexiones, (cota mínima de la función de evaluación) que nosotros obtenemos, es 2625 en lugar de 2620.

Número de elementos a colocar : 34

Número de elementos fijos : 0

Retícula : NC = 9; NF = 4

- Ejemplo 3 : IKER3

Número total de elementos a colocar = 56

Número de elemento fijos = 7

Retícula : NC = 8; NF = 7

Elementos fijos :

Elemento	Posición
50	49
51	41
52	33
53	25
54	17
55	9
56	1

- Ejemplo 4 : FLIFLO

Número de elementos a colocar = 64

Número de elementos fijos = 8

Retícula : NC = 8 ; NF = 8

Elementos fijos :

Elemento	Posición
57	64
58	56
59	48
60	40
61	32
62	24
63	16
64	8

- Ejemplo 5 : IL1

Este caso tomando de (10), procede de una tarjeta de la unidad de control del ordenador ILLIAC IV.

Número de elementos a colocar = 120

Número de elementos fijos = 15

Retícula : NC = 15 ; NF = 8

Elementos fijos :

Elemento	Posición
106	120
107	119
108	118
109	117
110	116
111	115
112	114
113	113
114	112
115	111
116	110
117	109
118	108
119	107
120	106

2.5.- Resultados obtenidos

Hemos utilizado los dos métodos de colocación inicial con los cinco ejemplos.

En cada caso, hemos medido el tiempo de ejecución t en segundos y hemos calculado la función de evaluación de la colocación obtenida $F(s)$.

La función de evaluación del problema de asignación - cuadrática, asociado a uno de colocación por la transformación especificada en el capítulo I es, evidentemente, para cualquier tipo de árbol:

$$F(s) = \frac{1}{2} \sum_{v_i, v_j \in V} c_{ij} d_{s(v_i), s(v_j)}$$

los resultados obtenidos, están resumidos a continuación, incluyendo en cada caso una cota mínima de la función de evaluación,

$$COTA = \frac{1}{2} \sum_{v_i, v_j \in V} c_{ij}$$

	t	F(s)
Colocación Aleatoria	2.7	419
Colocación Constructiva	1.1	350

Cota mínima = 160

33.-

	t	F(s)
Colocación Aleatoria	6.7	9850
Colocación Constructiva	6.5	6803

Cota mínima = 2625

Ejemplo 2 : STEIMBERG

	t	F(s)
Colocación Aleatoria	15.5	27384
Colocación Constructiva	22.7	23327

Cota mínima = 6053

Ejemplo 3 : IKER3

	t	F(s)
Colocación Aleatoria	19.7	4501
Colocación Constructiva	34.22	3227

Cota mínima = 915

Ejemplo 4 : FLIFLO

	t	F(s)
Colocación Aleatoria	68.7	7745
Colocación Constructiva	189.5	4536

Cota mínima = 1078

Ejemplo 5 : IL1

Estos resultados, ponen de manifiesto, algo que en principio es evidente: el método constructivo proporciona mejores resultados que el aleatorio. Sin embargo, no es para esto para lo que nosotros los hemos construido, sino para estudiar el comportamiento de los algoritmos iterativos partiendo de distintos métodos de colocación inicial.

De nuevo insistimos, en que los resultados del método constructivo pueden ser suficientemente buenos para algunas aplicaciones, y el tiempo de ejecución es bastante más pequeño que el que necesitarán los métodos iterativos.

Por último, a modo de ejemplo, incluimos la colocación obtenida por el método constructivo para el caso IL1.

COLOCACION

```

0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
0  28 74 84 77 40 37 23 19 11 8  12 34 43 11
0  57 73 51 76 47 35 10 6  12 4  13 44 7  48
55 73 71 58 60 51 20 14 26 9  3  21 27 95 3
88 83 39 57 53 53 24 87 16 2  49 22 23 89 93
100 81 70 62 54 51 64 66 80 1  25 19 29 94 96
101 77 60 83 67 52 63 68 79 23 46 50 96 92 91
120 116 118 117 119 115 114 113 112 111 110 109 108 107 106

```

C A P I T U L O I I I

EL PROBLEMA DE ASIGNACION CUADRATICA

Como ya hemos dicho antes, el problema de asignación cuadrática es un caso particular del problema de colocación. Sin embargo, este problema tiene una entidad y una importancia propias tales que merece ser definido por sí mismo.

El término "problema de asignación cuadrática" se -- debe a Koopmans y Beckmann (12) , que formularon un problema un poco más general del que vamos a definir aquí, en el campo de -- la Econometría.

3.1.- Definición formal del problema

Un grafo no orientado es un conjunto de elementos, de nominados vértices o nodos, V y un conjunto A de parejas de elementos, no ordenados, (v_i, v_j) denominados arcos, tales que $v_i, v_j \in V$.

Denominamos bucle a un arco del tipo (v_i, v_i) .

Un grafo no orientado, $G = (V, A)$, puede ser totalmente especificado por el conjunto de sus elementos $V = \{v_1, v_2, \dots, v_n\}$ y una matriz simétrica de orden n , $C = \{c_{ij}\}$, tal que

$$\begin{cases} c_{ij} = c_{ji} = 1 & \text{si } (v_i, v_j) \in A \quad \text{o} \quad (v_j, v_i) \in A \\ c_{ij} = c_{ji} = 0 & \text{si } (v_i, v_j) \notin A \quad \text{y} \quad (v_j, v_i) \notin A \end{cases}$$

A la matriz C se la denomina matriz de adyacencia -- del grafo.

Un grafo no orientado con pesos en sus arcos, es -- aquel que tiene una información numérica (peso) asociada a ca da arco. (En una definición más general la información asocia da a cada arco podrá ser alfanumérica). Un grafo de este tipo puede ser también especificado por el conjunto de sus elemen-- tos V , y su matriz de adyacencia, siendo $c_{ij} = c_{ji}$ el peso aso ciado al arco (v_i, v_j) o al arco (v_j, v_i) , en lugar de la uni-- dad.

Consideramos un grafo no orientado, con pesos en sus arcos y sin bucles, definido por el conjunto de sus n elemen-- tos $V = \{v_1, v_2, \dots, v_n\}$ y su matriz de adyacencia $C = \{c_{ij}\}$

Sea $P = \{p_1, p_2, \dots, p_r\}$ un conjunto de r puntos pertenecientes a un espacio métrico en el que está definida -- una distancia d , y tal que $r \geq n$.

Sea s una aplicación inyectiva del conjunto V en el conjunto P . Sea S el conjunto de todas las aplicaciones de es te tipo.

Para cada elemento $s \in S$ definimos una función de evaluación $F(s)$, de la siguiente manera:

$$F(s) = \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in V} c_{ij} d_{s(v_i), s(v_j)}.$$

El problema de asignación cuadrática, consiste en encontrar el elemento $s^* \in S$ tal que:

$$F(s^*) = \min \{ F(s) \mid s \in S \}$$

La definición formal que hemos presentado es la más adecuada para el problema, tal como se presenta en el diseño automático de sistemas digitales. Una generalización inmediata, puede hacerse sin más que considerar grafos orientados y hacer las correcciones adecuadas. (Por ejemplo la matriz de adyacencia C no sería simétrica).

A lo largo de este capítulo, aunque sería más adecuado usar el término asignación de los elementos de V a los puntos de P , seguiremos utilizando el término colocación por uniformidad de todo el trabajo.

3.2.- Antecedentes: Métodos conocidos más importantes

Además de los métodos constructivos ya tratados en -

el capítulo II, existen otros métodos para la solución del problema de asignación cuadrática que podemos clasificar en dos grupos:

- Métodos semi-enumerativos
- Métodos iterativos

3.2.1.- Métodos semienumerativos

Los métodos semienumerativos son los únicos conocidos capaces de encontrar la solución óptima del problema, siempre que el número de elementos que intervienen sea inferior al rango de 15 a 20 elementos. Por otra parte, permiten encontrar soluciones aproximadas para problemas de mayor número de elementos.

Estos métodos consisten esencialmente en ir construyendo de una manera ordenada, todas las posibles soluciones. En un estado determinado se dispone de la colocación de una parte de los elementos; a partir de esta colocación parcial se calcula una cota mínima de la función de evaluación de todas las posibles colocaciones totales que se derivan de ella. Esta cota se compara, con el valor de la función de evaluación de una posible solución al problema, que ha sido encontrada por cualquier método (por un método constructivo, o por el mismo algoritmo -- semi-enumerativo por ejemplo).

Si el valor de la cota es superior al de la solución disponible, se eliminan todas las colocaciones que se derivan de la colocación parcial disponible, y se cambia de posición, de forma ordenada, el último elemento colocado., reanudando el proceso.

En caso contrario se asigna, de forma ordenada, un nuevo elemento a la colocación parcial y se reanuda el proceso. Cuando se alcanza una colocación total, cuya función de evaluación tiene un valor inferior al de la posible solución, la colocación encontrada sustituye a la posible solución.

El proceso global, puede ser visualizado como la poda de un árbol de decisiones.

Evidentemente estos métodos conducen a la solución exacta del problema y también nos permiten encontrar soluciones aproximadas con un porcentaje determinado de error.

La eficacia de estos métodos, depende esencialmente de la bondad de la cota y de la posible solución utilizadas. Si no se utiliza ninguna cota tendríamos un método enumerativo. Por otro lado, cuanto mejor sea la cota más tiempo de cálculo necesita normalmente.

P.C. Gilmore (13), presenta dos cotas; un algoritmo óptimo, que él mismo considera no utilizable para más de 15 elementos y dos algoritmos subóptimos con tiempos de ejecución

creciendo con r^4 y r^5 .

E.L. Lawler (14), también utiliza estas técnicas y generaliza el problema a asignación cúbica y de orden N.

La necesidad de almacenar varias matrices $r \times r$ y el tiempo de ejecución que necesitan estos métodos, hacen -- que no sean adecuados para tratar problemas de tamaño elevado con un miniordenador.

3.2.2.- Métodos iterativos

Los métodos iterativos exigen como punto de partida, una colocación inicial de todos los elementos. Los principales métodos de este tipo son:

- Algoritmo de Steimberg
- Intercambio por parejas
- Métodos de relajación

Algoritmo de Steimberg

Este algoritmo (11), es el primer método importante, anterior a los trabajos de Gilmore y Lawler, para abordar el problema de asignación cuadrática, siendo superior a -- estos últimos en su capacidad para tratar problemas con un nú-

mero relativamente elevado de elementos.

Posteriormente R.A. Rutman (15), generalizó el método para tratar directamente el problema de colocación.

Esencialmente el algoritmo de Steimberg consiste - en la linealización del problema, por transformaciones sucesivas en problemas de asignación lineal.

El problema de asignación lineal, es una variante - del problema general de la programación lineal, que puede definirse de la siguiente forma:

Dada una matriz, $n \times n$, $C = \{c_{ij}\}$, encontrar una permutación p de los n primeros números enteros y positivos, para la cual se minimice

$$\sum_{i=1}^n c_i p(i).$$

Existen $n!$ posibles permutaciones, por lo que tampoco es posible aquí usar métodos enumerativos, sin embargo - existen métodos bien conocidos (16), (17) para resolver este problema, para n razonablemente elevado, del orden de - 200 elementos.

La idea básica para la linealización del problema consiste en, dada una colocación s , seleccionar un subconjunto de elementos de V , no conectados entre sí (entre los cuales no existe ningún arco), como candidatos para cambiar de posi-

ción. La asignación óptima de estos elementos, al conjunto de posiciones que han dejado libres y a las que estaban libres -- previamente si $r > n$, es un problema de asignación lineal.

Steimberg trabaja con subconjuntos de elementos de V no conectados entre sí que sean máximos, en el sentido de que no estén contenidos en ningún otro subconjunto de elementos de V no conectados entre sí.

El algoritmo de Steimberg, parte de una colocación inicial s_1 y de un conjunto $U = \{u_1, u_2, \dots, u_p\}$, tal que

$$\forall u_i \in U \quad \left\{ \begin{array}{l} u_i \subset V \\ \forall v_j, v_k \in u_i \quad c_{jk} = 0 \\ u_i \text{ es máximo} \end{array} \right.$$

A partir de s_1 y u_1 , por asignación lineal de los elementos u_1 a las posiciones libres de P obtenemos s_2 . Este -- proceso repetido para todos los elementos de U , constituye una iteración. El proceso se reanuda con u_1 y una nueva iteración. El algoritmo termina cuando en una iteración no se mejora el valor de la función de evaluación $F(s)$.

Este algoritmo depende esencialmente de tres factores:

- a) La colocación inicial s_1 utilizada.
- b) El conjunto U utilizado.

c) El orden en que se utilizan los elementos de U.

Intercambio por parejas

Para facilitar la exposición de este método, vamos a suponer que cuando el número de elementos n es inferior al número de puntos o posiciones r , el conjunto V se amplía con $r - n$ elementos ficticios $v_{n+1}, v_{n+2}, \dots, v_r$ tales que

$$C_{n+i, j} = 0 \quad \text{para} \quad \begin{cases} i = 1, \dots, r - n \\ j = 1, \dots, r \end{cases}$$

La idea básica de este método es muy sencilla. Dada una colocación inicial s_1 , una pareja de elementos de V intercambian sus posiciones. Se calcula el valor de la función de evaluación de la nueva colocación. Si este valor es inferior al correspondiente a la colocación antigua, se acepta el intercambio. En caso contrario, los elementos vuelven a sus posiciones primitivas. Este proceso repetido para todas las parejas de V constituye una iteración.

El algoritmo correspondiente termina cuando en una iteración no se mejora el valor de la función de evaluación, o cuando la mejora obtenida está por debajo de un porcentaje determinado del valor de la función de evaluación al comienzo de la iteración.

Si a la descripción anterior le añadimos la restricción, que no modifica el resultado final y economiza tiempo --

de cálculo, de que una pareja de elementos ficticios no intercambian sus posiciones, tenemos que el número de intentos de intercambio por iteración es:

$$NII = \frac{n(n-1)}{2} + n(r-n)$$

Esta idea es fácilmente generalizable a intercambios de varios elementos, sin embargo de las experiencias aportadas por Garside y Nicholson (18), se deduce, que la mejora de la función de evaluación es pequeña frente al intercambio por parejas y el tiempo de cálculo necesario mucho más elevado.

El comportamiento de un método basado en el intercambio por parejas, depende de dos factores:

- a) La colocación inicial utilizada
- b) El orden en el que se intercambian las parejas.

Métodos de relajación

Los métodos de relajación consideran el conjunto de elementos V , como un conjunto de partículas interactivas.

El primer método de este tipo se debe a Fisk, Caskey y West (19), que consideraron cuatro tipos de fuerzas actuando sobre cada elemento.

Partiendo de una colocación inicial, se calcula la fuerza resultante que actúa sobre cada elemento.

A cada elemento, secuencialmente, se le permite desplazarse en la dirección de su fuerza resultante, una distancia proporcional al módulo de su fuerza. El desplazamiento de todos los elementos constituye una iteración. El algoritmo correspondiente, termina cuando el desplazamiento medio de todos los elementos, es inferior a un valor predeterminado.

Un desarrollo posterior, basado en esta idea (6), considera los elementos como si estuvieran unidos por resortes, que cumplen la ley de Hooke $\vec{F} = K \vec{s}$.

La constante elástica del resorte que une los elementos v_i y v_j es c_{ij} .

Dada una colocación s , sobre cada elemento v_i actúa una fuerza resultante F_i tal que:

$$\vec{F}_i = \sum_{v_j \in V} c_{ij} \vec{d}_s(v_i, s(v_j))$$

Por otra parte, cada elemento tiene un punto de equilibrio perteneciente a R^2 , en el cual la fuerza resultante que actúa sobre él es nula.

Si llamamos p_i^* al punto de equilibrio de v_i , sus coordenadas pueden calcularse a partir de

$$\sum_{v_j \in V} c_{ij} \overrightarrow{d_{p_i^*, s(v_j)}} = 0$$

El punto de equilibrio, normalmente, no pertenecerá al conjunto de puntos P . Denominamos posición de equilibrio de v_i , $e(v_i) \in P$, al elemento de P más próximo a p_i^* .

A partir de estos conceptos, se pueden desarrollar varios métodos iterativos de relajación, que traten de llevar - secuencialmente cada módulo a su posición de equilibrio.

El estudio de la bibliografía disponible nos ha - llevado a la conclusión, de utilizar el intercambio por parejas como la técnica más adecuada para el tratamiento del problema de asignación cuadrática, en casos con un número elevado de elementos y utilizando un miniordenador como herramienta de trabajo.

Esta conclusión, está basada en las siguientes razones:

- a) La memoria necesaria es relativamente pequeña. La parte más importante, es la necesaria para el almacenamiento de la matriz de adyacencia C , que puede ser vectorizada.

- b) Creemos que la potenciabilidad que este método tiene, es adecuada para encontrar buenas soluciones, en problemas con un número elevado de elementos y en un tiempo razonable.
- c) Los resultados conocidos (5), (10), aunque evaluados directamente utilizando un árbol de expansión mínima, ponen de manifiesto un comportamiento del algoritmo de Steinberg - que, en el mejor de los casos, podemos considerar equivalente al de un intercambio por parejas con una selección prácticamente aleatoria de estas parejas.
- d) Los distintos factores que influyen en el comportamiento de este método no han sido suficientemente estudiados, un buen estudio de estos factores, llevará sin duda a mejoras en su comportamiento.
- e) Por último, la observación de que prácticamente todos los -- trabajos más recientes (3), (20), (21), (22), utilizan este método.

3.3.- Intercambio por parejas

Como ya hemos dicho antes, el comportamiento de - un método basado en el intercambio por parejas depende esencialmente de dos factores:

- La colocación inicial utilizada.

- El orden en el que se intercambian las parejas.

Para la colocación inicial, nosotros vamos a estudiar la influencia de una colocación inicial aleatoria y una colocación inicial constructiva.

El orden en el que se intercambian las parejas, puede ser descrito por unas normas, que construyan ordenadamente todas las posibles parejas. Unas normas genéricas pueden ser descritas de la siguiente manera:

- Seleccionamos, de acuerdo a algún criterio, un elemento $v_i \in V$, como elemento primario, tal que:

$$\sum_{v_j \in V} c_{ij} \neq 0$$

- v_i no ha sido previamente seleccionado como elemento primario en la iteración actual.
- El elemento v_i , se intercambia sucesivamente con el resto de los elementos de V , elementos secundarios, que no han sido previamente seleccionados como primarios, en la iteración actual.
- Dado un elemento primario v_i , todos sus posibles elementos secundarios se seleccionan ordenadamente, de acuerdo a algún criterio.

Quando no existen más elementos primarios, el proceso termina.

El orden en el que se intercambian las parejas, depende por lo tanto, del criterio para la selección del elemento primario y del criterio para la selección del elemento secundario.

Los criterios de selección de elemento primario y elemento secundario más utilizados, consisten en seleccionar los elementos en el mismo orden en el que se encuentran en el conjunto V. Muy similar a lo anterior, es seleccionarlos en un orden determinado por el conjunto numerado P, la única diferencia es que en este caso el orden cambiaría para cada iteración. Estos criterios son prácticamente aleatorios, en el sentido de que no podemos esperar que el ordenamiento de los elementos en V contenga ninguna información especialmente relevante.

Un criterio interesante, debido a R. H. Glaser - (23) , consiste en ordenar los elementos en sentido decreciente del valor de una función $Q(v_i)$ tal que:

$$Q(v_i) = \sum_{j \in V} c_{ij}$$

Los elementos primarios y secundarios se seleccionarían de acuerdo a este ordenamiento.

3.3.1.- Técnicas de relajación en el intercambio por parejas

M. Hanan y J. Kurtzberg (7), plantean un método, en nuestra opinión muy interesante, basado en aplicar

las ideas de los métodos de relajación al intercambio por parejas.

El objetivo fundamental, es tratar de reducir el número total de intentos de intercambio, que se producen en una iteración.

Para describir este método, vamos primero a definir el concepto de k -vecindad de un punto $p_i \in P$, que representaremos como $V_k(p_i)$.

Dados el conjunto de puntos P , un punto $p_i \in P$ y una constante k , definimos $V_k(p_i)$ como el subconjunto de puntos de P , tales que su distancia a p_i es menor o igual -- que k .

Recordemos por otra parte que la posición de equilibrio de un elemento v_i , la representamos como $e(v_i) \in P$.

Este método, consiste en aplicar al intercambio por parejas dos restricciones:

a) Sólo se intentan intercambiar aquellos elementos v_i, v_j , - tales que:

$$e(v_i) \in V_k(s(v_j))$$

$$e(v_j) \in V_k(s(v_i))$$

Esto produce que el número de intentos de intercambios por iteración, crezca linealmente con n . En efecto, este número está acotado por $2k(k+1)n$.

- b) Cuando se acepta el intercambio de dos elementos v_i, v_j , es tos elementos se asignan definitivamente a sus posiciones correspondientes, durante la iteración. Es decir, no vuelven a ser utilizados durante dicha iteración, ni como elementos primarios ni como secundarios.

El comportamiento de un método de este tipo, depende de los siguientes factores:

- Colocación inicial utilizada.
- Criterio de selección de elemento primario.
- Criterio de selección de elemento secundario.
- Valor del factor de vecindad k utilizado.

En nuestra opinión, este es el método con mayores posibilidades para el tratamiento de problemas con un número elevado de elementos.

La influencia de los distintos factores en su comportamiento, no ha sido suficientemente estudiada.

Tomando las ideas de este método como punto de partida hemos estudiado la influencia del factor de vecindad

en los resultados, hemos estudiado su comportamiento con distintos criterios de selección de elemento primario y secundario, así como la influencia de distintos tipos de colocación inicial. Hemos desarrollado una nueva estrategia de selección de elemento primario y secundario, que consideramos refuerza la potencialidad de este método. También hemos desarrollado otra estrategia, que determina el valor del factor de vecindad k , para cada iteración, en base a las características de la colocación disponible al comienzo de dicha iteración.

Por último, hemos comparado los resultados obtenidos con los proporcionados por distintos métodos de intercambio exhaustivo por parejas.

Es interesante destacar, que si suprimimos la restricción b), obtenemos un método intermedio entre éste y el intercambio exhaustivo por parejas. En este caso, dados una colocación inicial y unos criterios de selección de elemento primario y secundario, para $k = \infty$, tendríamos el método correspondiente a intercambio exhaustivo para estos factores. La influencia de la restricción a), puede ser aislada y analizada de esta manera.

3.4.- Una nueva estrategia

La idea básica que subyace en el método anterior, es la de reducir el número de intentos de intercambio, seleccionando, de una manera inteligente, únicamente aquellos que

tienden a llevar un elemento a una posición cercana a su punto de equilibrio.

Esta idea puede ser reforzada, con la estrategia que vamos a exponer a continuación para la selección del elemento secundario, que trata de llevar el elemento primario a la posición más próxima a su punto de equilibrio.

Dado un punto $p_i \in P$, con la definición de distancia que hemos dado, $V_k(p_i)$ es el conjunto de elementos de P , que están situados en el interior o en los lados del cuadrado de la figura 3.a.

Dados dos elementos v_i, v_j , la condición que deben cumplir para que se intente su intercambio, puede representarse gráficamente por la figura 3.b.

Esta condición puede expresarse y representarse de otra manera: dado un elemento primario v_i , cuya posición es $s(v_i)$ y cuya posición de equilibrio es $e(v_i)$, trazamos el cuadrado correspondiente con centro en $e(v_i)$, fig. 3.c. - El conjunto de todos los posibles elementos secundarios está formado, por los elementos situados en el interior o en los lados de dicho cuadrado y cuya posición de equilibrio cumple $e(v_j) \in V_k(s(v_i))$.

El criterio de selección de elemento secundario, que presentamos, se basa en la ordenación de todos los elementos secundarios en sentido creciente de su distancia a la po

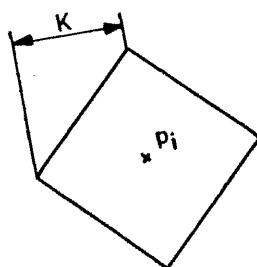


Fig.-3-a

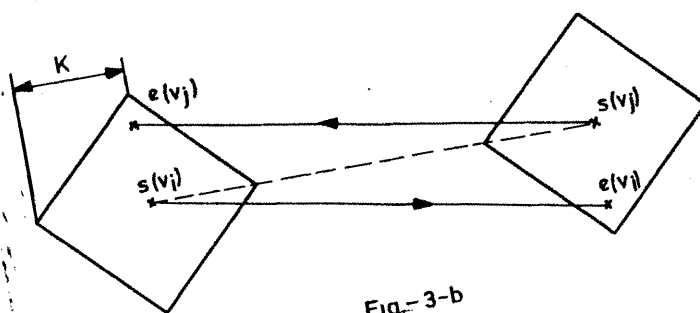


Fig.-3-b

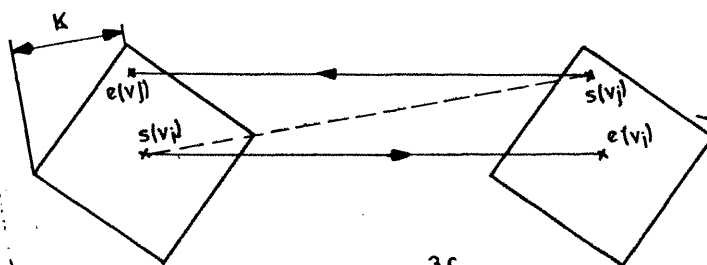


Fig.-3 c

Fig.-3

sición de equilibrio del elemento primario correspondiente.

La idea reflejada en la figura 3.c., nos permite ir obteniendo sucesivamente los elementos secundarios, a partir de la posición de equilibrio del elemento primario, sin necesidad de buscarlos todos y ordenarlos posteriormente. Esto es especialmente útil, porque en el momento en que se acepta un intercambio, no es necesario seguir utilizando los elementos secundarios restantes. El hecho de que los puntos de P sean los vértices de una retícula nos facilita este trabajo.

La rutina que hemos construido nos proporciona, para un punto p_i y una distancia d_0 , el conjunto de puntos de P que están situados a la distancia d_0 de p_i . Las llamadas sucesivas a esta rutina para valores crecientes de d_0 , nos proporcionan el ordenamiento deseado de los elementos secundarios sin más que mantener la tabla de referencias cruzadas adecuada.

Junto a este criterio para selección de elemento secundario, hemos utilizado un criterio para selección de elemento primario, basado en la ordenación de los elementos con sentido decreciente del valor $Q(v_i) = \sum_{v_j \in v} c_{ij}$.

Por último, hemos incluido la posibilidad de — que al comienzo de cada iteración, se calcule el valor de k a utilizar en base a las características de la colocación de — partida. El criterio que hemos seguido es el siguiente:

P1.- Para cada elemento $v_i \in V$, que no haya sido asignado a --
priori a una posición de una manera definitiva, se calcu-
la la distancia $d_{s(v_i), e(v_i)}$.

P2.- Hacemos k igual al máximo de todos los valores así calculados.

Este criterio produce normalmente unos valores elevados de k en las primeras iteraciones, que por otra parte, son las más rápidas, debido a que los intercambios son más numerosos y los elementos intercambiados no vuelven a intentar intercambiarse en una iteración.

A medida que la colocación disponible va mejorando, el valor de k disminuye.

En las últimas iteraciones, en las que la mejora de la función de evaluación es pequeña y el tiempo de ejecución más elevado el valor de k que se utiliza es pequeño.

Es interesante destacar que el valor de k así --
obtenido, es una cierta medida de la bondad de una colocación y del desequilibrio máximo de sus elementos.

3.5.- Cálculo incremental para la función de evaluación

Dada una colocación s_1 , cuando dos elementos v_1 y v_2 intercambian sus posiciones, obtenemos una nueva coloca-

ción s_2 .

El cambio producido en el valor de función de -
evaluación es $\Delta F = F(s_2) - F(s_1)$.

El intercambio por parejas se basa en aceptar -
los intercambios cuando $\Delta F < 0$.

La eficacia en tiempo de cálculo, de un método
basado en el intercambio por parejas, depende en gran parte -
de la manera de calcular F . La manera más costosa y más evi-
dente es, conocido $F(s_1)$, evaluar:

$$F(s_2) = \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in V} c_{ij} d_{s_2}(v_i), s_2(v_j),$$

y restarlos.

Para que un método de este tipo sea eficaz, es
imprescindible calcular directamente, el incremento producido
en el valor de la función de evaluación debido al intercambio
de dos elementos.

Para calcular directamente este valor, vamos a
denominar:

A_i : es el conjunto de elementos de V que están conectados a
 v_i . (0 lo que es equivalente, que tienen un arco en --
común con v_i).

L_i^1 : es la aportación al valor de la función de evaluación,
producida por el elemento v_i en la colocación s_1 .

L_i^2 : análogo a lo anterior, en la colocación s_2 .

En la colocación s_1 , tenemos que:

$$[1] \quad L_i^1 + L_j^1 = \sum_{v_k \in A_i} c_{ik} * d_{s_1(v_i), s_1(v_k)} + \\ + \sum_{v_k \in A_j} c_{jk} * d_{s_1(v_j), s_1(v_k)} - c_{ij} * d_{s_1(v_i), s_1(v_j)}$$

En la colocación s_2 , tenemos:

$$[2] \quad L_i^2 + L_j^2 = \sum_{v_k \in A_i} c_{ik} * d_{s_2(v_i), s_2(v_k)} + \\ + \sum_{v_k \in A_j} c_{jk} * d_{s_2(v_j), s_2(v_k)} - c_{ij} * d_{s_2(v_i), s_2(v_j)}.$$

Considerando que:

$$s_2(v_i) = s_1(v_j)$$

$$s_2(v_j) = s_1(v_i)$$

$$s_2(v_k) = s_1(v_k) \quad \text{para } v_k \neq v_i, v_j$$

$$v_j \in A_i \quad \text{y} \quad v_i \in A_j$$

Obtenemos:

$$\begin{aligned}
 [3] \quad & \sum_{v_k \in A_i} c_{ik} * d_{s_2(v_i), s_2(v_k)} = \\
 & = \sum_{v_k \in A_i} c_{ik} * d_{s_1(v_j), s_1(v_k)} + c_{ij} * d_{s_1(v_i), s_1(v_j)}
 \end{aligned}$$

$$\begin{aligned}
 [4] \quad & \sum_{v_k \in A_j} c_{jk} * d_{s_2(v_j), s_2(v_k)} = \\
 & = \sum_{v_k \in A_j} c_{jk} * d_{s_1(v_i), s_1(v_k)} + c_{ij} * d_{s_1(v_i), s_1(v_j)}
 \end{aligned}$$

Evidentemente, $\Delta F = (L_i^2 + L_j^2) - (L_i^1 + L_j^1)$

Realizando la resta, con los nuevos valores [3] y [4], sustituidos en [2] resulta:

$$\begin{aligned}
 [5] \quad \Delta F = & \sum_{v_k \in A_i} c_{ik} * \left(d_{s_1(v_j), s_1(v_k)} - d_{s_1(v_i), s_1(v_k)} \right) + \\
 & + \sum_{v_k \in A_j} c_{jk} * \left(d_{s_1(v_i), s_1(v_k)} - d_{s_1(v_j), s_1(v_k)} \right) + \\
 & 2 c_{ij} * d_{s_1(v_i), s_1(v_j)}
 \end{aligned}$$

A partir del resultado [5], es inmediata la construcción del algoritmo correspondiente, siendo de gran utilidad el mantener una tabla de referencias cruzadas entre elementos y posiciones.

Para analizar un intento de intercambio, los elementos se mantienen en sus posiciones primitivas s_1 ; sólo si $\Delta F < 0$, los elementos intercambian sus posiciones.

3.6.- Algoritmos contruidos para realizar un estudio comparativo

El comportamiento de una estrategia para resolver el problema de asignación cuadrática, no puede ser evaluado directamente por el grado de aproximación a la solución exacta del problema, puesto que, normalmente ésta no se conoce. Por ello, es siempre necesario compararla con otras estrategias ya existentes.

Los algoritmos que hemos construido pueden ser definidos y denominados de la siguiente manera:

A).- Numeramos los criterios de selección de elementos (3.3 y 3.4).

Tipo 1.- Criterio de selección aleatorio.

Tipo 2.- Criterio basado en $Q(v_i) = \sum_{\substack{j \\ v_j \in V}} c_{ij}$

Tipo 3.- Criterio de selección de elemento secundario -
propuesto en 3.4.

B) Utilizamos la letra "E" para los algoritmos exhaustivos; -
utilizamos la letra "R" para los algoritmos de relajación.

Los algoritmos que hemos construido son:

E11 : Algoritmo de intercambio exhaustivo por parejas, con -
criterios de selección de elemento primario y secunda-
rio del tipo 1.

E22 : Algoritmo de intercambio exhaustivo por parejas, con -
criterios de selección de elemento primario y secunda-
rio del tipo 2.

R11 : Algoritmo de relajación en el intercambio por parejas, .
con criterios de selección de elemento primario y secun
dario del tipo 1.

R22 : Algoritmo de relajación en el intercambio por parejas, -
con criterios de selección de elemento primario y secun-
dario del tipo 2.

R23 : Algoritmo de relajación en el intercambio por parejas, -
con criterio de selección de elemento primario del tipo
2 y criterio de selección de elemento secundario del ti-
po 3.

R23^{*} : El mismo que el anterior, pero con ajuste automático -
del factor de vecindad según el criterio expuesto en -
3.4.

Todos ellos son capaces de tratar problemas que
tengan hasta 150 elementos.

Estos algoritmos los hemos aplicado a los cinco
ejemplos descritos en el capítulo II, en cada caso a las dos -
colocaciones iniciales, aleatoria y constructiva.

3.7.- Fluctuación de los resultados con el factor de vecindad

Como ya se ha dicho en 3.3.1., el comportamien-
to de los algoritmos de relajación en el intercambio por pare-
jas, depende del factor de vecindad k utilizado.

En la figura 4., pueden verse los resultados ob-
tenidos con el ejemplo nº 2, partiendo de su colocación inicial
aleatoria, para los distintos valores de k .

En esta figura, puede observarse que el valor mí-
nimo de la función de evaluación la alcanza el algoritmo R23 -
con $k = 8$. Este valor, 5003, es todavía superior al que alcanza
el algoritmo R23^{*} directamente: 4986.

Un estudio similar, realizado con el ejemplo nº3

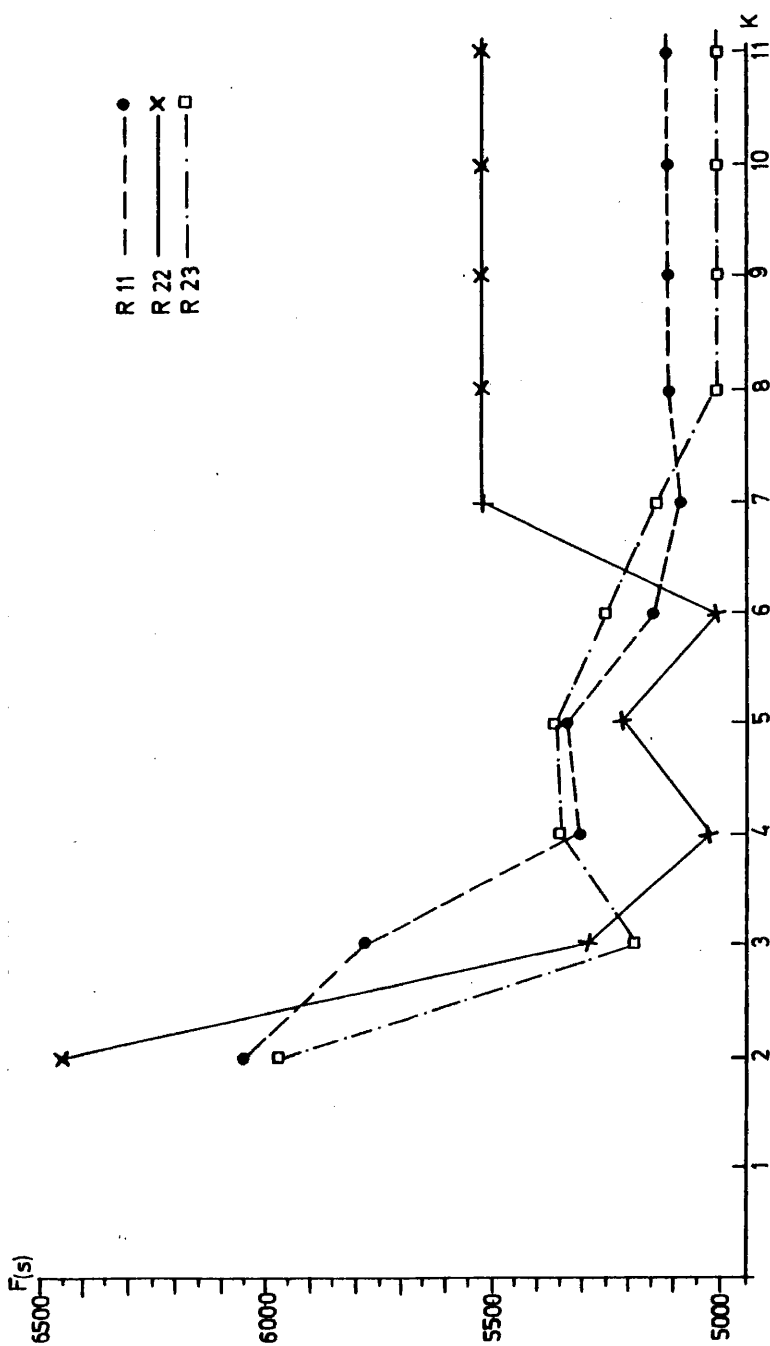


Fig.-4

partiendo también de su colocación inicial aleatoria, conduce a un valor mínimo de $F(s)$ de 22056, obtenido con el algoritmo R23 y $k = 7$. El valor proporcionado por $R23^*$ directamente es 22074.

Analogamente, el estudio del ejemplo nº 4, conduce a un valor mínimo de $F(s)$ de 2593, obtenido con el algoritmo R23 y $k = 7$. El valor proporcionado por $R23^*$ es 2599.

El comportamiento del nuevo algoritmo $R23^*$ en los tres casos es muy satisfactorio, y por otra parte es de destacar que los valores mínimos, se alcanzan con uno de los dos algoritmos nuevos: R23 o $R23^*$.

Hemos realizado un estudio similar, partiendo de la colocación inicial constructiva, con los ejemplos nº 2 y nº 4.

En el ejemplo nº 2, el valor mínimo, 5287, lo alcanza R11 con $k = 5$. En este caso $R23^*$ proporciona 5543.

En el ejemplo nº 4, el valor mínimo, 2684, lo alcanza R23 con $k = 5$. $R23^*$ proporciona 2744.

En general, los algoritmos de relajación se han comportado peor partiendo de una colocación inicial constructiva que de una colocación inicial aleatoria. El comportamiento de $R23^*$ no es tan correcto en el caso de una colocación inicial constructiva.

Con una sola excepción, R23 ha proporcionado el mínimo absoluto de todos los valores obtenidos.

Conociendo estas fluctuaciones, si se dispone de tiempo de ordenador suficiente, se puede obtener una solución muy buena con un "barrido" de este tipo para los distintos valores de k . En todos los casos que hemos estudiado, esta solución es mejor que la proporcionada por los algoritmos exhaustivos E11 y E22.

Sin embargo, si no se dispone de este tiempo, se puede buscar una solución con un algoritmo como R23^M, o bien, con un algoritmo de relajación tipo R11, R22 o R23, fijando a priori un valor de k .

El valor $k = 5$, que utiliza M. Hanan, es el - que hemos usado para comparar los algoritmos R11 y R22, con el nuevo R23.

Los resultados obtenidos los exponemos a continuación.

3.8.- Resultados obtenidos: Analisis y Conclusiones

Las tablas nº 1, nº 2, nº 3, nº 4 y nº 5, resumen los resultados obtenidos por los distintos algoritmos.

En cada caso $F(s)$ representa el valor final alcanzado por la función de evaluación y t el tiempo de ejecución, en un miniordenador PDP 11-40 con dedicación absoluta de la CPU a estas tareas. (Es decir, durante su ejecución no se -

	C I A L		C I C O	
	F(s)	t	F(s)	t
Coloca ción - Inicial	419	2.7	350	1.1
E11	322	14.0	320	7.1
E22	308 [*]	14.0	309 ⁺	10.6
R11	334	16.1	324	10.7
R22	314	21.9	325	9.0
R23	309 ⁺	13.8	325	9.4
R23 [*]	309 ⁺	11.5	325	8.5

Cota minima de F(s) = 160

Ejemplo nº 1 : UCA

T A B L A N° 1

	C I A L		C I C O	
	F(s)	t	F(s)	t
Coloca- ción Ini- cial	9850	6.7	6803	6.5
E11	5077 ⁺	189.8	5532	95.6
E22	5104	127.0	5248	127.5
R11	5332	232.1	5287	159.1
R22	5210	129.2	5422	164.0
R23	5343	127.9	5543	170.6
R23 [*]	4986 [*]	150.9	5543	164.0

Cota minima de F(s) = 2625

Ejemplo nº 2 : STEIMBERG

T A B L A N° 2

	C I A L		C I C O	
	F(s)	t	F(s)	t
Coloca- ción Inicial.	27384	15.5	23327	22.7
E11	22198	906.2	22225	389.6
E22	22160	518.9	22240	389.9
R11	22242	511.7	22224	762.0
R22	22082	922.1	22223	765.1
R23	22060 [*]	548.6	22245	869.8
R23 [*]	22074 ⁺	895.1	22260	934.2

Cota minima de F(s) = 6053

Ejemplo nº 3 : IKER3

T A B L A N º 3

	C I A L		C I C O	
	F(s)	t	F(s)	t
Coloca- ción In- cial.	4501	19.7	3227	34.22
E11	2763	505.4	2707	505.6
E22	2745	505.7	2701	506.3
R11	2633 ⁺	634.1	2714	687.1
R22	2703	508.0	2697	523.5
R23	2647	520.7	2684	451.4
R23 ^x	2599 ^x	628.2	2744	403.1

Cota minima de F(s) = 915

Ejemplo nº 4 : FLIFLO

T A B L A N º 4

70.-

	C I A L		C I C O	
	F(s)	t	F(s)	t
Coloca- ción In- cial.	7745	68.7	4536	189.5
E11	3900	3287.0	3712	3943.0
E22	3901	4600.2	3754	3944.0
R11	3741	2591.0	3752	1311.9
R22	3756	2216.1	3725	1187.3
R23	3630 [*]	1753.8	3758	1385.4
R23 [*]	3705 ⁺	4624.0	3733	1908.2

Cota minima de F(s) = 1078

Ejemplo nº 5 : IL1

T A B L A N° 5

ha permitido la multiprogramación).

El encabezamiento de las tablas, CIAL y CICO, indica los resultados obtenidos a partir de la colocación -- inicial aleatoria y constructiva respectivamente.

Los algoritmos R11, R22 y R23 han trabajado -- con un factor de vecindad $k = 5$.

La primera fila de cada tabla, representa los valores de la función de evaluación de la colocación inicial correspondiente, ya obtenidos en el capítulo II, indican por tanto el valor de partida para los distintos algoritmos.

En la parte inferior de cada tabla se indica -- el valor de una cota inferior de $F(s)$, calculada a partir -- del valor : $\frac{1}{2} \sum_{v_i, v_j \in V} c_{ij}$.

En las distintas tablas hemos señalado con un asterisco el caso en el que la función de evaluación alcanza el valor mínimo de todos los disponibles. Hemos señalado con una cruz, el valor siguiente.

El análisis de estos resultados, pone de manifiesto los siguientes aspectos:

- 1.- En general los distintos algoritmos, con algunas excepciones, llegan a mejores resultados finales partiendo de la colocación inicial aleatoria que partiendo de la colocación inicial constructiva, aunque con un tiempo de ejecución superior.
- 2.- Los algoritmos de relajación conducen a mejores resultados que los exhaustivos y en un tiempo de ejecución inferior. Esto se pone sobre todo de manifiesto en el ejemplo nº 5, es decir cuando el número de elementos (120) es elevado.
- 3.- Entre los nuevos algoritmos se alcanza el valor mínimo en los cuatro ejemplos de mayor tamaño y además alcanzan el siguiente valor, más próximo al mínimo, en tres de los cinco ejemplos.
- 4.- El comportamiento del nuevo algoritmo R23^M, alcanza siempre el mínimo o el segundo valor en todos los casos, partiendo de una colocación inicial aleatoria. Sin embargo el tiempo de ejecución es bastante elevado cuando trabaja con el ejemplo nº 5.
- 5.- El nuevo algoritmo R23 tiene generalmente un comportamiento mejor que los otros algoritmos de relajación R11 y R22.
- 6.- Consideramos que, globalmente, tomando en cuenta el tiempo de ejecución y el valor de la función de evaluación, R23 es el algoritmo más estable y la mejor elección.

C A P I T U L O I V

EL PROBLEMA DE COLOCACION

El problema de colocación ya ha sido definido formalmente en el capítulo I, donde se han concretado algunos aspectos, tales como el espacio métrico utilizado R^2 y la definición de distancia más adecuada, $d_{ij} = |x_i - x_j| + |y_i - y_j|$.

En el capítulo II se ha establecido que, para el estudio del problema de colocación a nivel abstracto, el conjunto de puntos (o posiciones) P , está formado por los vértices de una retícula cuadrangular de lado unidad.

También en el capítulo I, se han definido los principales tipos de árboles adecuados al diseño automático de sistemas digitales. Las normas que actualmente sigue un diseñador manual, hacen que el tipo de árbol que más se aproxima al problema real sea el árbol Steiner. El algoritmo que disponemos para trazado de conexiones, capaz de unir un punto a un conjunto de puntos previamente conectados, también es un argumento a favor del árbol Steiner. Sin embargo, la gran dificultad para determinar la longitud del árbol Steiner, que une un conjunto de puntos, hace que el árbol de expansión mínima sea un substituto adecuado. De hecho, la diferencia entre las longitudes del

árbol de expansión mínima y del árbol Steiner, que unen un conjunto de puntos, se va haciendo menos importante a medida que aumenta el número de puntos.

En este capítulo nos planteamos el problema de colocación, utilizando el árbol de expansión mínima como tipo de árbol T^* que une un conjunto de puntos.

4.1.- Métodos iterativos para tratar directamente el problema de colocación

Algunos de los métodos utilizados para el tratamiento del problema de asignación cuadrática, descritos en 3.2., pueden ser utilizados para tratar el problema de colocación -- sin más que utilizar la función de evaluación definida en el capítulo I. La diferencia fundamental es que esta función, necesita un tiempo de cálculo mucho más elevado que el que necesita la función de evaluación del problema de asignación cuadrática.

No existen métodos semienumerativos, conocidos, para resolver el problema de colocación. Esto se debe a la dificultad de encontrar un buen procedimiento para calcular una cota inferior para todas las colocaciones totales que se derivan de una colocación parcial dada.

El algoritmo de Steimberg es un método válido. El primero en utilizarlo para el problema de colocación fué R.A. Rutman.

Los métodos basados en el intercambio por parejas, son evidentemente utilizables.

Los métodos de relajación pueden utilizarse, siempre que se disponga de memoria suficiente para almacenar simultáneamente:

- a) la información específica del problema de colocación, necesaria para calcular la función de evaluación.
- b) la información correspondiente a un problema de asignación cuadrática asociado al de colocación, necesaria para calcular la fuerza resultante sobre cada elemento, o su punto de equilibrio.

En nuestro caso, trabajando con un miniordenador, - hemos rechazado los métodos de relajación por economía de memoria. No utilizamos el algoritmo de Steimberg por las mismas razones expuestas en 3.2. Vamos a utilizar, por lo tanto, un método basado en el intercambio exhaustivo por parejas para - el tratamiento directo del problema de colocación.

Por otra parte, es éste el método más utilizado para este problema, usando criterios de selección de elemento - primario y secundario prácticamente aleatorios.

Es importante señalar que se ha construido este algoritmo, con la finalidad de comparar su comportamiento, con el de la transformación del problema de colocación en asigna-

ción cuadrática que proponemos.

4.1.1.- Algoritmo utilizado.

El algoritmo que utilizamos para tratar directamente el problema de colocación, se basa en el intercambio exhaustivo por parejas con criterios de selección de elemento primario y secundario del tipo 1.

El tiempo de ejecución, previsiblemente alto, que - va a necesitar, ha hecho que procuremos optimizar la construcción de este algoritmo, adoptando las siguientes medidas:

- a) Analizamos, al comienzo, el conjunto de conexiones C , buscando conexiones iguales entre sí (es decir, compuestas -- por los mismos elementos). Eliminamos todas las conexiones repetidas de C , reduciendo por tanto su cardinal, y establecemos una información asociada a cada uno de los elementos restantes de C : su índice de repetición.
- b) Creamos una tabla de referencias cruzadas entre elementos y conexiones. En cada momento podemos obtener así, de una manera rápida, los elementos que componen una conexión dada y las conexiones a las que pertenece un elemento dado.
- c) Dada una colocación s , para cada elemento $c_1 \in C$, mantenemos como información asociada, la longitud de los árboles - de expansión mínima que unen las posiciones de los elementos de c_1 .

- d) Utilizamos un cálculo incremental para la función de evaluación, basado en el siguiente razonamiento:

Sean v_i, v_j los elementos que intentan intercambiar sus posiciones en un momento determinado. Consideremos una conexión cualquiera $c_1 \in C$.

- Si $v_i, v_j \in c_1$, la longitud del árbol de expansión mínima, que une las posiciones de los elementos de c_1 , $T_z^*(c_1)$, no varía con el intercambio.
- Si $v_i, v_j \notin c_1$, tampoco varía $T_z^*(c_1)$, con el intercambio.
- Únicamente si, $v_i \in c_1$ y $v_j \notin c_1$

$$v_i \notin c_1 \text{ y } v_j \in c_1$$

La longitud $T_z^*(c_1)$ puede variar con el intercambio.

Por lo tanto, cuando dos elementos intentan intercambiar sus posiciones, sólo evaluamos la longitud de los árboles de expansión mínima correspondientes a las conexiones de este último tipo. Su comparación con las longitudes correspondientes, disponibles de la colocación anterior, nos permitirán decidir si aceptamos o rechazamos el intercambio.

Este conjunto de medidas implica utilizar más memoria con objeto de reducir el tiempo de ejecución. El algorit-

mo, tal como se ha construido, puede tratar problemas con un máximo de 150 elementos y 300 conexiones.

Para el cálculo de la longitud del árbol de expansión mínima, que une un conjunto de puntos K, hemos utilizado el siguiente método:

Consideremos dos conjuntos A y B. El conjunto A contendrá en cada instante los puntos, ya conectados por ramas - del árbol de expansión mínima, el conjunto B contendrá los no conectados.

P1.- [Inicializa] $A = \emptyset, \quad B = K$

P2.- [Selecciona primer punto] . Selecciona arbitrariamente - un punto de B, suprimiéndolo de B e incluyendolo en A.

P3.- [Determina la siguiente
rama del árbol.] Selecciona el punto contenido en B, que este más próximo a alguno de los puntos de A. El punto de B así seleccionado y el punto de A al que esta más próximo, determinan la siguiente rama del árbol.

Actualizar A y B, suprimiendo el punto correspondiente de B e incluyendo en A.

Actualizar la longitud del árbol, sumándole la longitud de esta rama.

P4.- [Test de terminación.] Si $B = \emptyset$, el algoritmo termina.
En caso contrario ir a P3.

4.2.- Métodos iterativos para tratar el problema de colocación por transformación en asignación cuadrática.

Como ya se ha explicado en el capítulo I, a todo problema de colocación se le pueden asociar varios problemas de asignación cuadrática, dependiendo de la función de transformación que se adopte.

Una vez transformado, el problema de asignación cuadrática puede ser tratado por cualquiera de los métodos expuestos en 3.2. La solución aproximada de este último problema, es también una solución aproximada para el problema de colocación.

La ventaja de este enfoque es, evidentemente, la mayor rapidez con que se calcula la función de evaluación del problema de asignación cuadrática.

El principal inconveniente reside en que, una buena solución, del problema de asignación cuadrática asociado, no tiene por qué ser una buena solución para el problema de colocación original.

Su eficacia depende esencialmente, en nuestra opinión, de la función de transformación que se adopte. Es esta función de transformación, la que debe conseguir que una buena solución al problema de asignación cuadrática, lo sea también para el problema de colocación.

En nuestra opinión, éste es el enfoque más correcto para tratar problemas con un número elevado de elementos, y pensamos que su importancia irá en aumento a medida que crezca el tamaño de los problemas a resolver.

El estudio comparativo realizado en el capítulo III, entre diversos algoritmos, para el problema de asignación cuadrática, nos lleva a utilizar el algoritmo R23 con $K=5$ (para problemas del tamaño que estamos estudiando), como la mejor opción, cualquiera que sea la función de transformación adoptada.

En la transformación de un problema en otro, el conjunto V de elementos y el conjunto P de puntos o posiciones, permanecen comunes a ambos.

Unicamente es necesario definir, como se construye la matriz de adyacencia (o de interconexiones) $C^* = \{c_{ij}\}$ del

problema de asignación cuadrática a partir de la información contenida en el conjunto de conexiones $C = \{c_1, c_2, \dots, c_m\}$ del problema de colocación.

4.2.1. Transformación clásica

La transformación más sencilla y más utilizada, es la siguiente:

$$[1] \quad c_{ij} = \sum_{c_1 \in C} w_1 \quad \text{siendo} \quad \begin{cases} w_1 = 1 & \text{si } v_i, v_j \in c_1 \\ w_1 = 0 & \text{en caso contrario.} \end{cases}$$

4.2.2. Una transformación nueva.

Sea una conexión c , formada por t elementos. La transformación [1], supone que existe un arco (o una interconexión), entre cada pareja de elementos de c . El número de arcos obtenidos de esta manera es:

$$(t-1) + (t-2) + \dots + 2 + 1 = \frac{t(t-1)}{2}$$

Sin embargo, el número de arcos necesario para unir los t elementos de la conexión c es $t-1$.

Supongamos que la distancia entre cada pareja de posiciones, correspondientes a los elementos de c , es la misma y

la denominamos d^* . (Se pueden encontrar ejemplos de esta situación para dos, tres, y cuatro puntos).

La aportación de la conexión al valor de la función de evaluación, del problema de asignación cuadrática asociado, a través de la transformación [1], es en estas condiciones:

$$\frac{t(t-1)}{2} d^*$$

Por otra parte, la aportación de la conexión c al valor de la función de evaluación, del problema de colocación es: $(t-1) d^*$

Si quisieramos que en estas condiciones, el problema de asignación cuadrática reflejara exactamente el problema de colocación, deberíamos utilizar en [1], para la conexión c , un valor $W = 2/t$.

Esta situación que acabamos de describir, refleja exactamente la realidad cuando $t = 2$ y en algunos casos de tres y cuatro elementos. Es una buena aproximación cuando el número de elementos es pequeño y sus posiciones están concentradas, es decir, cuando la distancia media entre todas las posibles parejas de posiciones, se aproxima a la distancia media entre las parejas de posiciones que definen el árbol de expansión mínima.

La situación opuesta se produce cuando el número de elementos que intervienen es elevado y sus posiciones están dispersas.

En este caso la distancia media entre todas las posibles parejas de posiciones, va siendo cada vez más elevada - (a medida que crece el número de elementos) que la distancia media entre las parejas de posiciones que definen el árbol de expansión mínima. Un caso típico se produce cuando las posiciones de los t elementos están alineadas y separadas por una distancia d_0 .

En estas condiciones, la suma de las distancias entre cada pareja de posiciones es:

$$\sum_{i=1}^{t-1} \sum_{j>i} d(p_i, p_j) = \sum_{i=1}^{t-1} \frac{(i+1) i}{2} \cdot d_0$$

La distancia media entre dos posiciones es:

$$\bar{d} = \frac{\sum_{i=1}^{t-1} \frac{(i+1) i}{2} \cdot d_0}{\frac{t(t-1)}{2}}$$

Realizando las operaciones necesarias, teniendo en cuenta que

$$\sum_{i=1}^{I=N} i^2 = \frac{N^3}{3} + \frac{N^2}{2} + \frac{N}{6}$$

Obtenemos:

$$\bar{d} = \frac{t+1}{3} d_0$$

Esta distancia media \bar{d} , crece linealmente con el número de elementos, manteniéndose siempre mayor que la distancia media entre las parejas de posiciones que definen el árbol de expansión mínima, que en este caso coincide con d_0 .

Considerando que generalmente, las conexiones con un número de elementos superior a diez, son bastante escasas, podemos tomar como valor medio, representativo de esta situación el de $t = 5$, resultando $\bar{d} = 2d_0$.

En este caso, para que la aportación de la conexión al problema de asignación cuadrática, fuera la misma que su aportación al problema de colocación tendríamos que usar $W = 1/t$ para esta conexión.

En efecto:

$$\frac{1}{t} \frac{t(t-1)}{2} d_0 = (t-1) d_0$$

Basándonos en estos razonamientos, hemos decidido adoptar un valor W , dependiente del número de elementos de cada conexión, que sea el promedio de las dos situaciones anteriores.

Es decir : $w = \frac{2}{t} + \frac{1}{t} / 2 = \frac{3}{2t}$

Utilizaremos por tanto, la siguiente transformación:

$$[2] \quad c_{ij} = \sum_{c_1 \in C} w_1 \quad \text{siendo} \quad \begin{cases} w_1 = \frac{3}{2 \cdot |c_1|} & \text{si } v_i, v_j \in c_1 \\ w_1 = 0 & \text{en caso contrario} \end{cases}$$

en donde representamos por $|c_1|$ el cardinal del conjunto c_1 .

4.3. Ejemplos estudiados y resultados obtenidos.

Los ejemplos estudiados proceden de los casos reales de diseño de tarjetas de circuitos impresos, descritos en el apéndice I.

Cada tarjeta da origen a un problema de colocación, que queda descrito en el apéndice II.

El conjunto de puntos, o posiciones, P que utilizamos para cada caso, está definido por la retícula correspondiente, ya descrita en el capítulo II.

Hemos tratado cada uno de estos ejemplos con tres procedimientos; el algoritmo descrito en 4.1.1 para resolución directa, el algoritmo R23 con $K = 5$ usando la transformación

[1] , y el algoritmo R23 con $K = 5$ usando la nueva transformación [2] .

Para cada ejemplo se ha trabajado con las colocaciones iniciales aleatoria y constructiva obtenidas en el capítulo I.

Las tablas nº 6, nº 7, nº 8 y nº 9 resumen los resultados obtenidos, utilizando un formato similar al usado en el capítulo II.

$F(s)$ indica en cada caso el valor final alcanzado por la función de evaluación, calculado en base a los árboles de expansión mínima correspondientes.

El tiempo de ejecución t , ha sido medido, también en estos ejemplos, con dedicación exclusiva de la CPU a las correspondientes tareas.

Analizando estos resultados, se pueden extraer las siguientes conclusiones:

- 1º) El tiempo de ejecución para el algoritmo de resolución directa es bastante más pequeño, cuando parte de una colocación inicial constructiva que cuando parte de una aleatoria. Sin embargo el resultado final es en unos casos mejor y en otros del mismo orden, partiendo de la colocación inicial aleatoria.

Este fenómeno se pone sobre todo de manifiesto en el ejemplo nº 2, que contiene un grupo numeroso de conexiones con un número elevado de elementos.

	C I A L		C I C O	
	F(s)	t	F(s)	t
Colocación Inicial	253	2.7	201	1.1
Resolución Directa	181	95.0	179	26.0
Transformación Clasica 1	181	14.0	190	9.5
Transformación Nueva 2	181	15.62	179	18.91

Ejemplo nº 1 : UCA

T A B L A N º 6

	C I A L		C I C O	
	F(s)	t	F(s)	t
Colocación Inicial	872	15.5	790	22.7
Resolución Directa	638	6584	686	3271
Transformación Clásica 1	769	551.9	789	869.8
Transformación Nueva 2	707	609.0	716	790.52

Ejemplo nº 2 : IKER 3

T A B L A N º : 7

	C I A L		C I C O	
	F(s)	t	F(s)	t
Colocación Inicial	1369	19.7	1063	34.22
Resolución Directa	806	1653	802	1182
Transformación Clasica 1	957	514.6	977	453.2
Transformación Nueva 2	888	441.1	798	419.8

Ejemplo nº 3 : FLIFLO

T A B L A N º : 8

		C I A L		C I C O	
		F(s)	t	F(s)	t
Colocación		2925	68.7	1775	189.5
Inicial					
Resolución					
Directa		1361	3783	1389	2844
Transformación					
Clasica	1	1420	1764	1467	1384
Transformación					
Nueva	2	1350	1640	1348	1917

Ejemplo nº 4 : IL1

T A B L A N º : 9

2º) La transformación clásica conduce, prácticamente en todos - los casos, a resultados finales más elevados que los obtenidos por resolución directa. Sin embargo los tiempos de ejecución son en todos los casos notablemente inferiores, oscilando entre la mitad y la décima parte.

En este caso, siempre se obtiene el mejor resultado final cuando se parte de la colocación inicial aleatoria.

3º) La transformación nueva [2] , que hemos propuesto, conduce siempre a mejores resultados finales que la transformación clásica [1] . Los tiempos de ejecución considerados en conjunto son prácticamente equivalentes para todos los casos.

4º) Si comparamos los resultados proporcionados por la transformación nueva [2] , con los obtenidos, por resolución directa, vemos que con una sola excepción (ejemplo nº 2), la -- transformación nueva obtiene siempre el mejor resultado de todos los disponibles y con un tiempo de ejecución oscilando entre la mitad y la décima parte de los correspondientes a la resolución directa.

El comportamiento, no tan correcto, de [2] con el ejemplo nº 2 puede ser explicado en base a la cantidad anormalmente alta de conexiones con un número elevado de elementos.

5º) En resumen, creemos que la transformación que hemos presentado conduce a resultados finales muy buenos con un tiempo de ejecución razonablemente bajo.

C A P I T U L O V

EL PROBLEMA DE COLOCACION EN EL DISEÑO AUTOMATICO DE TARJETAS DE CIRCUITOS IMPRESOS

En los capítulos anteriores, se ha definido y se han planteado nuevos métodos para el tratamiento del problema de colocación, a nivel abstracto, en el sentido de asignar elementos a posiciones.

En este capítulo, se aborda el problema práctico, de colocación óptima de componentes en el diseño automático de tarjetas de circuitos impresos.

En el tratamiento del problema real se plantean fundamentalmente tres cuestiones.

- a) Qué entendemos por colocación óptima?
- b) Cómo introducir en la función de evaluación los factores a - optimizar.
- c) Cómo pasar de la información diversificada del problema real, a las estructuras de datos adecuadas para utilizar los métodos del problema abstracto.

Para la primera cuestión, coherentemente con nuestros objetivos, consideramos como colocación óptima, aquella que cum
ple los siguientes requisitos, enunciados en orden de prioridad absoluta:

- 1º Permitir al algoritmo de trazado de conexiones, la unión de todos los puntos a conectar.
- 2º Minimizar el número de taladros, o cambios de cara, necesarios.
- 3º Minimizar la longitud total de conexionado.
- 4º Minimizar el tiempo de ejecución del algoritmo de trazado - de conexiones.

La eficacia de una colocación, expresada en estos -- términos, sólo puede ser determinada por el propio algoritmo - de conexionado. Ya hemos dicho, a lo largo de esta memoria que el algoritmo de colocación y el algoritmo de conexionado deben estar estrechamente relacionados, debiendo depender el tipo de árbol que se use, del algoritmo de conexionado de que se dis
ponga, siempre que se cumplan las restricciones tecnológicas.

Con este fin, se ha preparado una variante del progra
ma que ejecuta el algoritmo de conexionado, que sin dibujar nada por el trazador de curvas, mida en su ejecución exclusivamente el número total de puntos a conectar, el número total de pun
tos no conectados, el número de taladros necesarios, la longitud total de conexionado y el tiempo de ejecución.

Las medidas realizadas para los distintos ejemplos, se estudian en el apartado 5.3. de este capítulo.

La función de evaluación definitiva que hemos adoptado, se explica en el apartado 5.2.

A continuación analizamos el tratamiento de la información procedente del problema real.

5.1. Lenguaje orientado al problema

Un conjunto de trabajos, realizados anteriormente en el campo del diseño automático de sistemas digitales, (2), (24), (25), (26), condujeron a la definición y utilización de un lenguaje de descripción de tarjetas de circuitos impresos, como el medio más eficaz de entrada de datos al algoritmo de conexiónado de que disponemos.

Este lenguaje es también el soporte de entrada de datos para el algoritmo de colocación. La información básica sobre los elementos que intervienen, el tipo de elemento y las conexiones que entre ellos deben realizarse es la misma para el algoritmo de conexiónado y el de colocación.

La primera tarea a realizar fué, por lo tanto, modificar el traductor de dicho lenguaje, para que generara las estructuras básicas de datos adecuadas al problema de colocación o al de asignación cuadrática, según el tratamiento:

a) Tabla de elementos y lista de conexiones.

b) Tabla de elementos y matriz de interconexiones.

La matriz de interconexiones (o de adyacencia), se vectoriza en base a la transformación:

$$c^x(i, j) = v(k)$$

$$\text{Siendo: } k = \frac{(i-1) \times (i-2)}{2} + j \quad \text{para } i > j$$

Por otra parte, es necesario proporcionar al nuevo traductor una información suplementaria, específica del problema de colocación, que el trazado automático de conexiones no necesitaba. Esta información suplementaria es de dos tipos: de definición del problema y de definición de ligaduras o restricciones que el algoritmo de colocación debe ser capaz de manejar.

Para la definición del problema, el espacio físico de una tarjeta de circuitos impresos, se convierte a priori y manualmente en una reticula rectangular, de tal forma que cada uno de los rectangulos elementales de dicha reticula sea capaz de contener a cualquiera de los elementos que van a ser colocados. Los centros de dichos rectangulos definen los puntos p_i , del conjunto P de posibles posiciones. Es necesario aclarar que un elemento del problema abstracto de colocación, puede corresponder a uno a varios componentes del problema práctico, pudiéndose agrupar varios componentes discretos entre si o con elementos integrados para definir un elemento del problema abstracto.

La primera información que hay que proporcionar al algoritmo de colocación, es por lo tanto la descripción de dicha retícula, es decir su número de filas y su número de columnas.

El producto del número de filas por el número de columnas define el número total de posiciones r . El algoritmo de colocación trabajará con un número de elementos igual al número de posiciones, introduciendo si es necesario elementos ficticios no conectados a ningún otro elemento.

Estas retículas, aunque no están dibujadas como tales, pueden apreciarse en las figuras nº 6, nº 9, nº 12, nº 15 correspondientes a las cuatro tarjetas estudiadas.

Las ligaduras o restricciones que el traductor, en su estado actual, es capaz de manipular son los siguientes:

- a) Una pastilla de circuitos integrados puede ser asignada, de forma definitiva, a una posición.
- b) Uno o varios componentes discretos pueden ser asignados, de forma definitiva, a una posición, incluso si ésta ya tiene asignada una pastilla. Por ejemplo, en las figuras nº 6, -- nº 7 y nº 8, las resistencias R_1 , R_2 , R_3 , R_4 , R_5 y los condensadores C_1 y C_2 están asignados siempre a la posición número 18.
- c) Un componente ficticio, no conectado a ningún otro, puede -

ser asignado de forma definitiva a una posición, manteniéndole definitivamente libre. Esto es necesario para proteger zonas prohibidas, utilizadas para anclajes por ejemplo, de la asignación de elementos.

- d) El tratamiento de los conectores, o conjuntos de terminales, es un aspecto fundamental para un buen programa de colocación automática.

Cada conector da lugar a uno o varios elementos del problema abstracto, dependiendo de su geometría y de la geometría de la retícula, que serán asignados de forma definitiva a sus posiciones correspondiente.

En las figuras nº 6, nº 7 y nº 8 los conectores son - ESA y ESB; en las figuras nº 9, nº 10 y nº 11, los conectores son D e I que se superponen. En las figuras nº 12, nº 13 y nº 14 los conectores son CONA, CONC, CONB, COND.

Estos conectores pueden apreciarse mejor en las figuras nº 17, nº 18, nº 19 y nº 20.

Para cada conector se define su nombre y orientación, la posición de la primera pata, el número de patas asignadas a la primera posición y el número de patas que se asigna a cada una de las restantes posiciones que pueda ocupar.

El traductor definirá como definitivamente ocupadas, las posiciones a las que se haya asignado algún elemento y ha-

r  intervenir estos componentes en la generaci n de la lista de conexiones y de la matriz de interconexiones.

Modificando ligeramente el traductor ser  posible en el futuro introducir nuevas ligaduras, si se cree conveniente, para un mejor tratamiento de los elementos discretos en tarjetas anal gico-digitales, conservando el algoritmo de colocaci n, sin modificar, como n cleo fundamental del sistema.

5.2. Disponibilidad de pistas: Una nueva definici n de distancia.

El problema de colocaci n, a nivel abstracto, ha sido definido con una funci n de evaluaci n basada fundamentalmente en el tipo de  rbol que une un conjunto de puntos.

Ya se ha razonado, en el cap tulo IV, la elecci n -- del  rbol de expansi n m nima, como el m s adecuado al problema real de dise o autom tico de sistemas digitales, desde un punto de vista pr ctico.

En principio, parece que ser  suficiente sustituir la ret cula cuadrangular de lado unidad, utilizada en el problema de colocaci n a nivel abstracto, por una ret cula rectangular de lados DX, DY definidos por la geometr a de la tarjeta, para pasar a las caracter sticas del problema real.

Sin embargo pensamos, y los resultados nos lo han ra

tificado, que la minimización de la longitud geométrica de conexión, no es una meta que resuma los objetivos que definen una colocación óptima, en el sentido dado al comienzo de este capítulo.

La idea fundamental que aportamos, consiste en dar un contenido a la definición de distancia, superior al meramente geométrico, incluyendo en ella una información sobre disponibilidad de pistas y dificultades de trazado de conexiones.

Consideremos la figura 5.a., que representa cuatro circuitos integrados de 16 patas equidistantes, contenidos en una tarjeta de 24 x 16 décimas de pulgada. La retícula correspondiente, de dos filas y dos columnas, quedaría definida por $DX = 11$ y $DY = 7$.

Si la función de evaluación se define en base a la distancia geométrica, el problema de colocación trataría de situar las pastillas más conectadas en vertical y las menos conectadas en horizontal. Con esto, no habríamos facilitado el trazado de conexiones, sino por el contrario lo habríamos entorpecido.

En efecto, si suponemos que pueden pasar dos pistas por cada décima de pulgada y suponemos prohibido el borde útil, a 0.05" del borde geométrico, el número total de pistas horizontales disponibles es 25 y el número total de pistas verticales disponibles es 29, debido a las pistas que inutilizan las patas de los circuitos integrados.

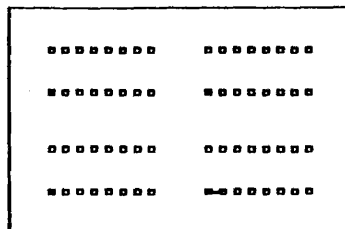


Fig. 5.a

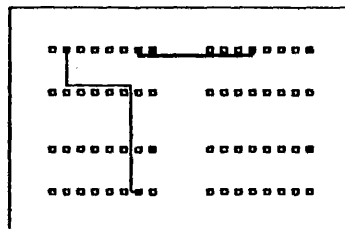


Fig. 5.b

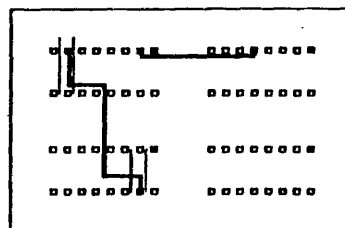


Fig. 5.c

Esta proporción 29 a 25 es un indicador, si tratamos de facilitar el conexionado, mucho mejor que la proporción de distancias 11 a 7.

Si usamos $DX = 11$ y $DY = 7$ en el programa de colocación, conseguiremos que existan un gran número de conexiones -- en vertical con una proporción pequeña de pistas disponibles para realizar el conexionado, entorpeciendo realmente.

Por ello, en una primera aproximación al problema, la mejor solución sería la que hemos adoptado de partida, haciendo $DX = DY = 1$, que normalmente es un indicador equilibrado de la disponibilidad de pistas, en horizontal y vertical.

Sin embargo, todavía no es éste un buen reflejo de la realidad. Consideremos la fig. 5.b en la que se dibujan una conexión en horizontal y una conexión en vertical. La conexión en horizontal utiliza una sola pista (horizontal) y probablemente no necesitará ningún taladro. La conexión en vertical utiliza dos pistas (una horizontal y otra vertical) y necesita un taladro seguro y probablemente dos.

La situación para la conexión vertical es todavía -- peor en la figura 3.c, por estar ocupadas las dos pistas adyacentes a la pata de partida.

En resumen, las conexiones en vertical son más costosas que las conexiones en horizontal, en una proporción como -- mínimo de dos a uno por lo que atenerse a las distancias geomé-

tricas desvirtua el problema real.

Por lo tanto, considerando el eje de abcisas coincidente con el lado alargado de las pastillas, para dar a la distancia vertical un valor p veces superior al horizontal, nos basta con establecer, para el problema de colocación, una definición de distancia:

$$d_{ij} = |x_i - x_j| + p |y_i - y_j|$$

El algoritmo desarrollado utiliza $p = 3$, para el -- cálculo de la función de evaluación.

En resumen, el algoritmo de colocación trabaja con la transformación en asignación cuadrática presentada en 4.2.4, utiliza el algoritmo R23, con $k = 5$, presentado en 3.4 para la resolución del problema de asignación cuadrática, y usa la definición de distancia que se ha establecido en este apartado, para el cálculo de la función de evaluación.

5.3. Comparación de las colocaciones obtenidas por distintos métodos.

Tal como hemos planteado al comienzo de este capítulo, la eficiencia de una colocación debe ser medida, en última instancia, por los resultados que permite obtener al algoritmo de conexionado.

Con objeto de realizar un estudio comparativo entre distintos algoritmos, se han utilizado las cuatro tarjetas descritas en el apéndice I. Las tarjetas IKER3 y FLIFLO han sido seleccionadas entre las tarjetas procedentes de distintos usuarios que han presentado mayores dificultades para su resolución.

En este estudio se han utilizado tres métodos de colocación:

1ª Colocación manual proporcionada, en las tarjetas UCA, IKER3 y FLIFLO, por el usuario tras varios días (excepto en UCA) de estudio del problema. No se dispone de colocación manual para la tarjeta IL1.

La colocación manual de las tarjetas UCA, IKER3 y FLIFLO está representada en las figuras nº 6, nº 9, y nº 12, añadiendo a su nombre la terminación "OR".

2ª Colocación obtenida por el algoritmo de resolución directa, descrito en 4.1.1, seleccionando la que proporciona un valor final menor de la función de evaluación, cuando se parte de colocaciones iniciales constructiva y aleatoria. La definición de distancia que se utiliza es la rectilínea normal dada en el capítulo I.

El tiempo empleado para obtener esta colocación, es mucho más elevado que el necesario para obtener una colocación del tercer tipo. (Véanse las tablas nº 6, nº 7, nº 8 y nº 9.).

104

23-MAY-79

11:59:31

UCAOR TAR

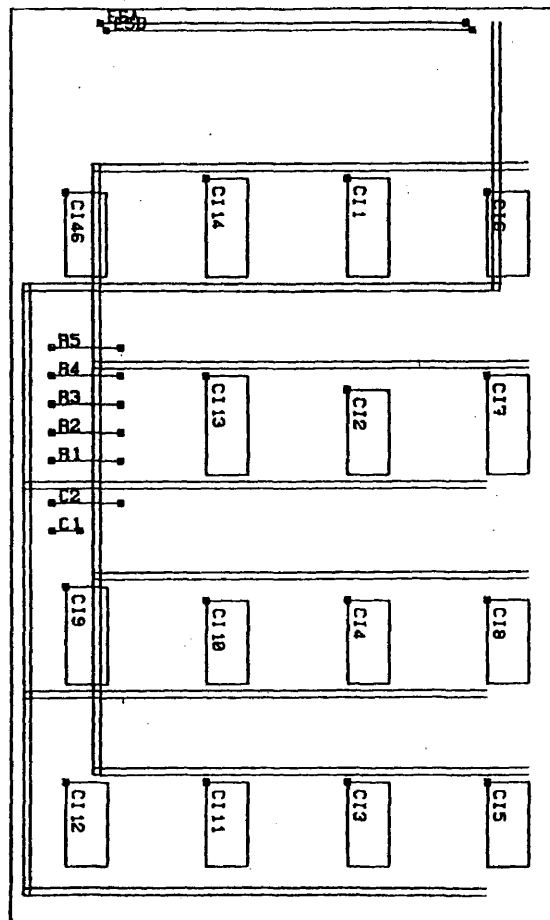


FIGURA Nº 6

105

23-MAY-79

12-82-12

UCAST . TAR

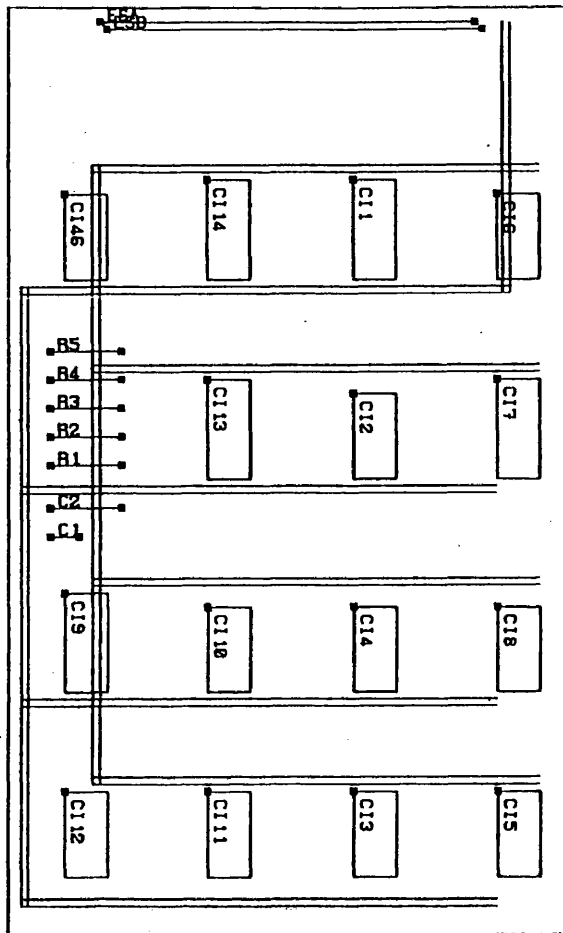


FIGURA Nº 7

UCADV.TAR

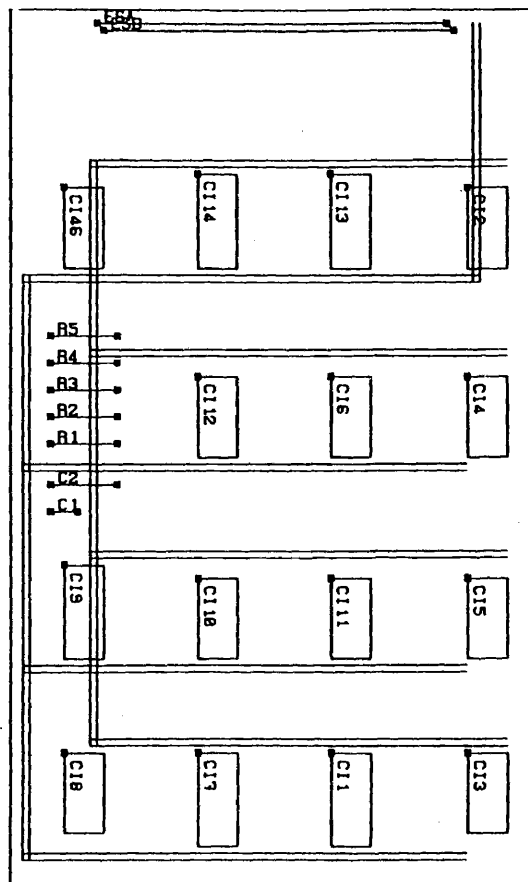


FIGURA Nº 8

23-MAY-79

11:41:04

IKER30R.TAR

107

U1	U3	U5	U7	U9	U11	U13
U2	U4	U6	U8	U10	U12	U14
U33	U37	U41	U45	U47	U13	U14
U34	U38	U42			U15	U16
U35	U39	U43			U17	U18
U36	U40	U44	U46	U48	U29	U31
U19	U21	U23	U25	U27	U29	U31
U20	U22	U24	U26	U28	U30	U32

23-MAY-79

11.36.39

IKER3ST.TAR

108

U41	U46	U6	U24	U23	U31	U21
U47	U49	U48	U12	U32	U22	U27
U45	U14	U8	U28	U42	U38	U29
U48	U18	U43	U38	U39	U26	U25
U36	U35	U37	U44	U5	U2	U1
U33	U28	U16	U13	U11	U9	U7
U34	U19	U17	U15	U4	U18	U3

23-MAY-79

10-08-05

IKER3DV.TAR

109

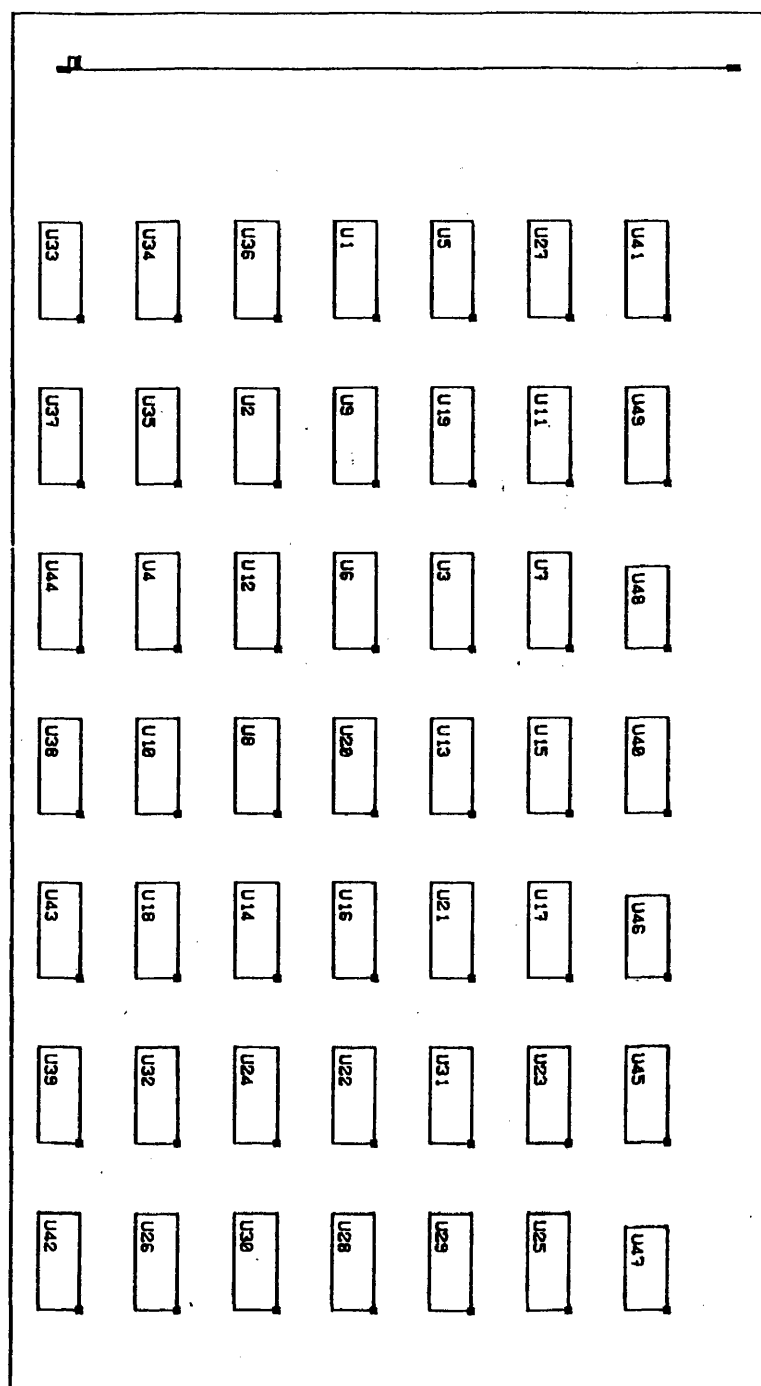


FIGURA Nº 11

23-MAY-79

12.07.49

FLIFLOOR.TAR

110

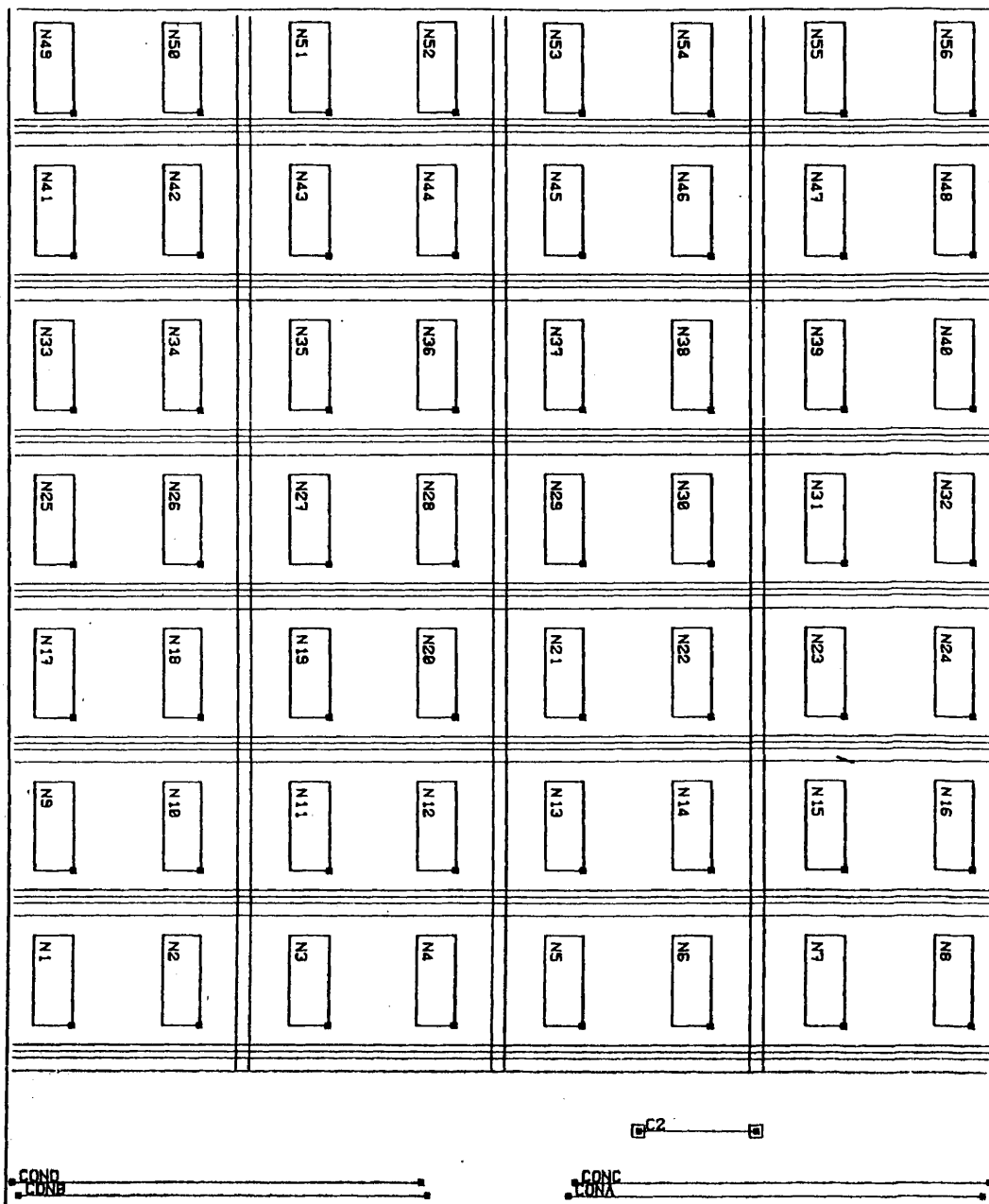


FIGURA Nº 12

23-MAY-79

12.14.16

FLIFLOST TAR

///

N43	N44	N48	N49	N51	N55	N56	N47
N41	N42	N34	N39	N54	N5	N32	N58
N52	N31	N3	N48	N26	N27	N29	N25
N37	N38	N33	N18	N28	N21	N24	N38
N1	N46	N17	N19	N36	N13	N15	N14
N28	N45	N18	N4	N35	N22	N23	N8
N9	N2	N11	N5	N12	N53	N7	N16

C2



FIGURA N° 13

BIBLIOTECA

23-MAY-79

12:21:26

FLIFLODV.TAR

112

N49	N52	N48	N58	N1	N41	N43
N42	N18	N37	N33	N38	N5	N2
N35	N34	N36	N39	N18	N11	N3
N12	N48	N17	N19	N28	N4	N5
N28	N26	N31	N24	N23	N21	N53
N27	N32	N29	N25	N22	N6	N13
N54	N45	N46	N38	N15	N14	N7
N55	N47	N56	N51	N8	N44	N16

2

COND
COND

COND
COND

FIGURA N° 14

AS7	AS6	AS1	A180	A75	A37	A48	A39	A17	A20	A21	A14		A25	A2
AS7	AS9	AS9	AS5	AS3	A36	A38	A15	A18	A24	A15	A5	A11	A25	A2
AS5	AS7	A72	A73	A8	A1	A23	A4	A48	A25	A22	A8	A18	A5	A3
AS8	A71	A78	A85	A84	A19	A30	A44	A49	A42	A45	A9	A7	A41	A49
AS6	AS4	A77	A50	AS6	AS3	AS9	AS4	A27	A45	A28	A25	A13	AS5	
AS3	AS8	A74	AS1	AS9	AS2	AS1	AS2	AS3	AS6	A47	A4	AS6	AS5	AS2
AS7	AS9	AS9	AS7	AS6	AS5	A79	AS4	A78	AS6	A30	A31	A12	A23	AS1
AS4	AS1	AS2	AS3	AS4	AS5	AS6	AS7	AS8	AS9	AS10	AS11	AS12	AS13	AS14

FIG. N° 15

19-03-50

IL 10V. TAR

11/1

A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A59	A57	A56	A59	A81	A85	A7	A5	A4	A45	A44	A42	A43	A41	A36	
A56	A61	A75	A3	A2	A18	A11	A8	A25	A47	A12	A33	A6	A37		
A69	A97	A83	A13	A9	A24	A25	A22	A21	A18	A24	A22	A17	A16		
A176	A58	A87	A54	A53	A50	A38	A25	A27	A49	A46	A48	A26	A28	A38	
A75	A77	A52	A72	A73	A8	A1	A23	A28	A21	A18	A24	A22	A17	A16	
A89	A71	A78	A68	A69	A63	A62	A86	A19	A15	A84	A88	A91	A95	A93	
A55	A53	A52	A57	A52	A65	A64	A79	A78	A80	A84	A88	A91	A95	A93	
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63	A64	A79	A78	A80	A84	A88	A91	A95	A93
A180	A174	A51	A65	A67	A62	A63									

FIGURA Nº 16

Esta colocación está representada, para las distintas tarjetas, por las figuras nº 7, nº 10, nº 13 y nº 15, en las que se ha añadido al nombre de la tarjeta la terminación "ST".

3ª Colocación proporcionada por el nuevo algoritmo, tal como - ha quedado descrito al final de 5.2., seleccionando la que proporciona un valor final menor de la función de evaluación, cuando se parte de colocaciones iniciales constructiva y aleatoria.

Esta colocación está representada, para las distintas tarjetas, por las figuras nº 8, nº 11, nº 14 y nº 16, - en las que se ha añadido al nombre de la tarjeta la terminación "DV".

Cada tarjeta, con cada una de las colocaciones descritas, se ha procesado con una variante del programa que ejecuta el algoritmo de conexionado, con objeto de medir en cada caso:

- a) Número total de puntos a conectar: NTP
- b) Número total de puntos que no se han podido conectar: --
NTPN
- c) El número de taladros, o cambios de cara necesarios: NT
- d) La longitud total de conexionado: LON
- e) El tiempo de ejecución: TIM

Nombre	NTP	NTPNC	NT	LØN	TIM
UCAOR	237	0	163	6614	8' 46''
UCAST	237	0	140	5774	5' 48''
UCADV	237	0	144	5959	7' 11''
IKER3QR	678	26	773	19209	8h 28' 19''
IKER3ST	678	22	608	16337	3h 28' 1''
IKER3DV	678	4	739	16799	1h 10' 56''
FLIFLOOR	669	22	652	27916	5h 24' 24''
FLIFLOST	669	41	635	28719	4h 2' 13''
FLIFLODV	669	27	615	27912	4h 38' 11''
IL1ST	839	29	932	26930	6h 12' 25''
IL1DV	839	26	886	27092	5h 45' 47''

T A B L A N° 10

Los resultados obtenidos están resumidos en la tabla nº 10, donde los tres tipos de colocación se identifican por las terminaciones "OR", "ST" y "DV" antes mencionados.

Estos resultados ponen de manifiesto que el algoritmo de colocación que hemos desarrollado, proporciona unas colocaciones mejores que las manuales en todos los parámetros medidos, con la única excepción de FLIFLODV en la que deja 5 puntos más sin conectar.

Así mismo el nuevo algoritmo proporciona unos resultados globalmente muy superiores a los proporcionados por el método clásico.

Incluimos por último, las figuras nº 14, nº 18, nº 19, y nº 20 que representan el diseño obtenido por el sistema para cada tarjeta, utilizando el algoritmo de colocación desarrollado y el trazado automático de conexiones.

118

6 MAY-79

11:07:30

UCADV TAR

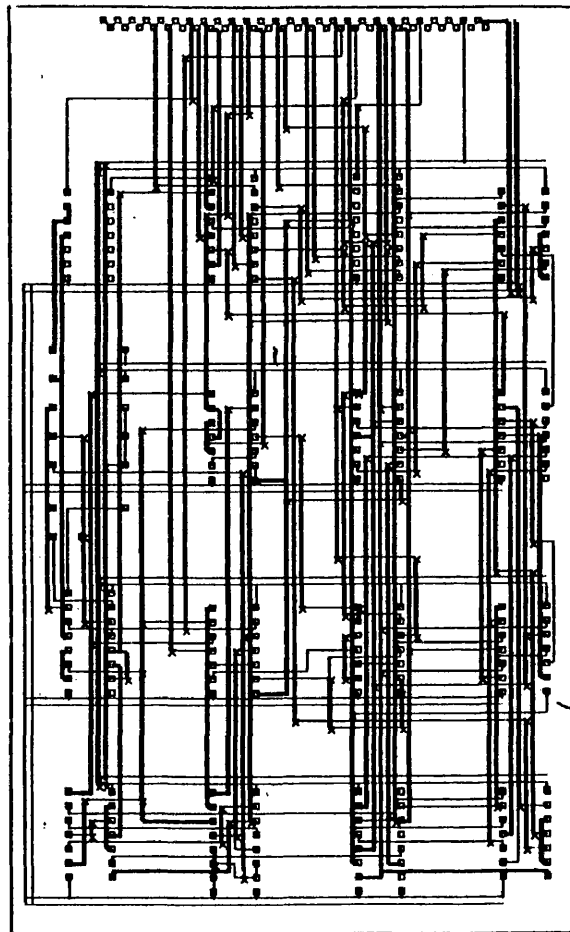


FIGURA Nº 17

25-MAY-79

11:28:19

IKER3DV.TAR

119

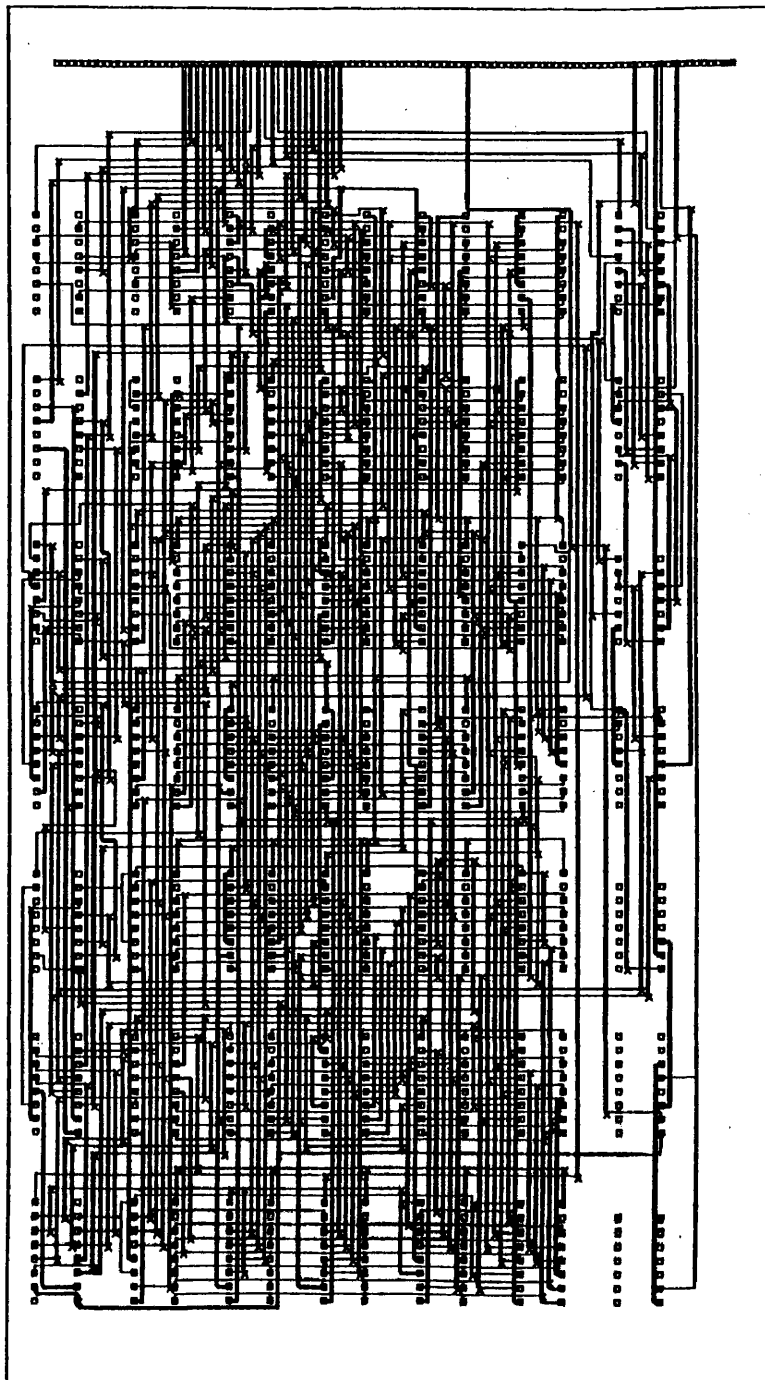


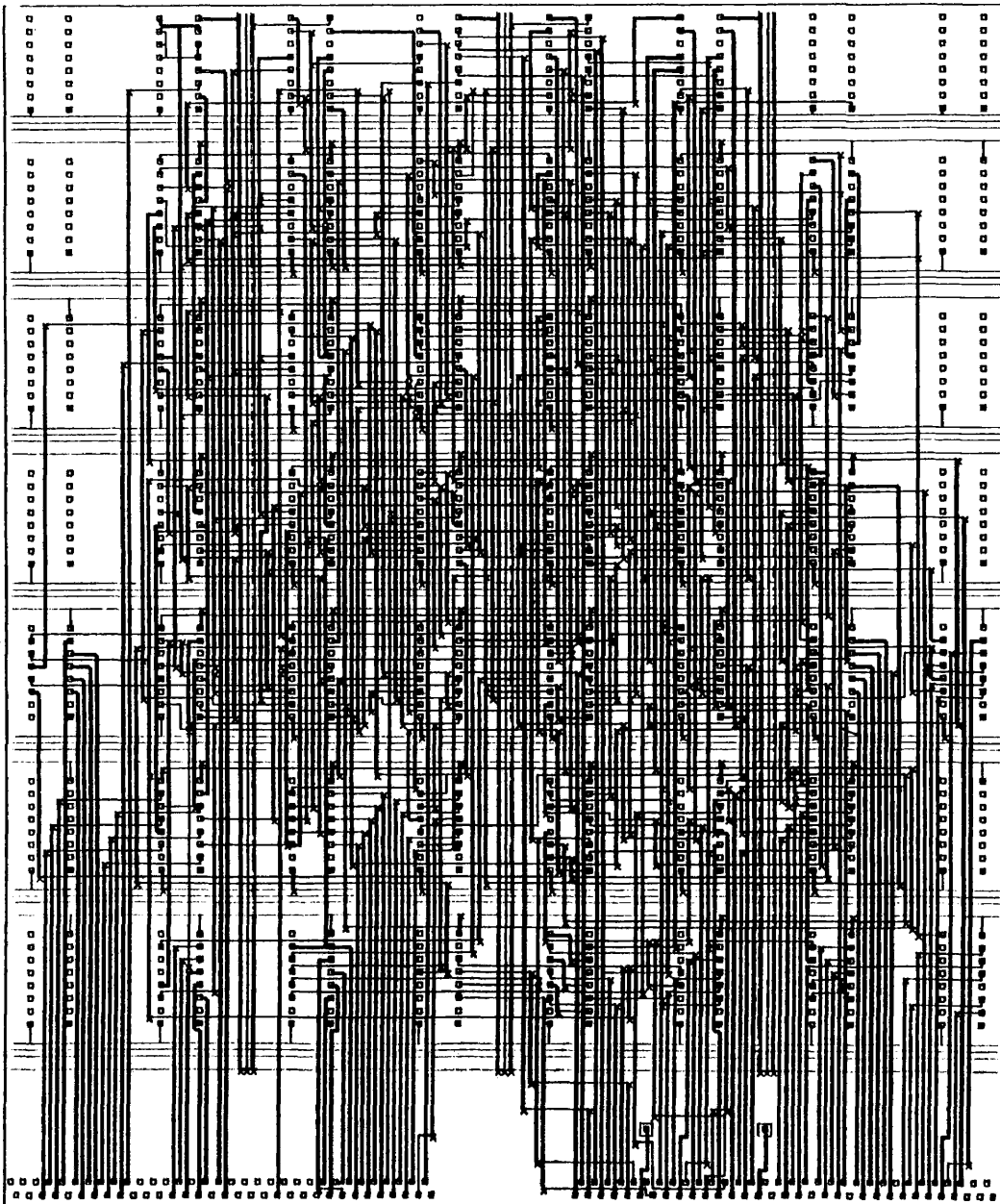
FIGURA Nº 18

25-MAY-79

16.26.13

FLIFLODV TAR

FIGURA Nº 19



26-MAY-78

17-28-13

IL 10V TAR

121

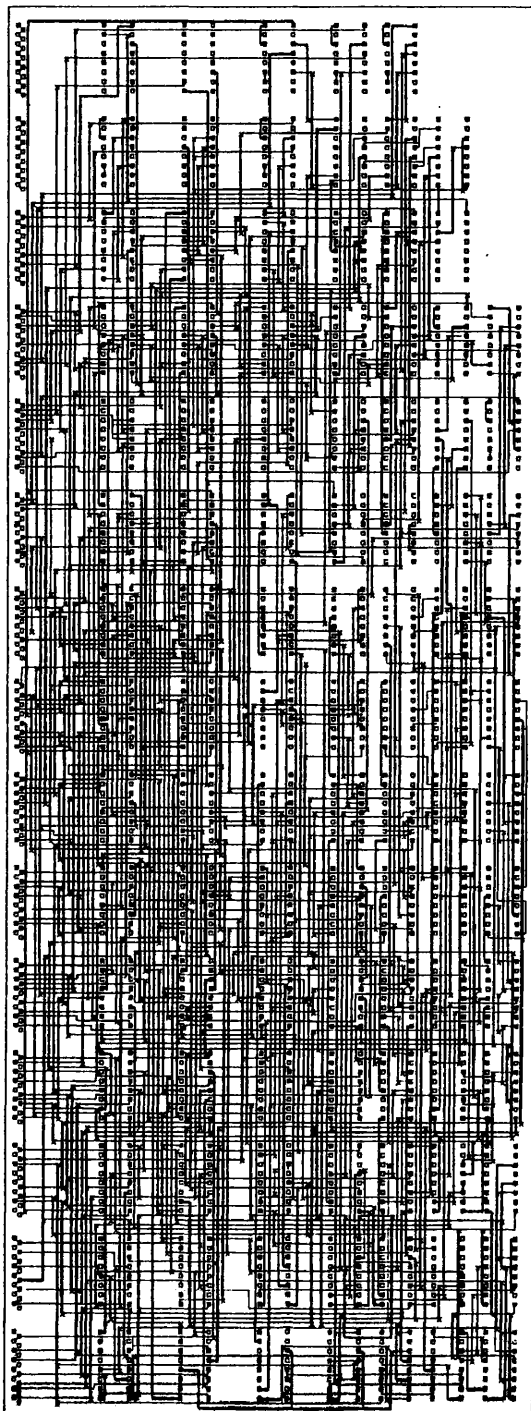


FIGURA Nº 20

PRINCIPALES APORTACIONES Y CONCLUSIONES

1.- Se ha presentado una definición formal y genérica del problema de colocación, válida para diversas aplicaciones, utilizando una función de evaluación basada en el tipo de árbol que une un conjunto de puntos. Posteriormente, se ha adoptado el árbol de expansión mínima como el más adecuado, desde el punto de vista práctico, al diseño automático de sistemas digitales.

2.- Se ha enfocado el problema de colocación, descomponiéndolo en dos partes:

- a) Encontrar un buen algoritmo para resolver el problema de asignación cuadrática.
- b) Encontrar una buena transformación del problema de colocación en un problema de asignación cuadrática asociado.

Este enfoque ha demostrado su eficacia, al reducir el tiempo de ejecución y el valor final alcanzado por la función de evaluación, como puede verse en las tablas nº 6, nº 8 y nº 9.

3.- Se ha presentado un nuevo algoritmo para el problema de asignación cuadrática, basado en la aplicación de técnicas de relajación al intercambio por parejas, en el que se ha introducido un criterio de selección de elemento secundario

que refuerza la heurística básica de relajación y un método para ajuste automático del factor de vecindad.

- 4.- Se ha realizado un estudio comparativo del comportamiento del nuevo algoritmo frente al de otros algoritmos conocidos, en la resolución del problema de asignación cuadrática, utilizando los grafos procedentes de los ejemplos de diseño de tarjetas de circuitos impresos estudiados.

Este estudio pone de manifiesto que el nuevo algoritmo es el más estable de todos ellos y proporciona generalmente los mejores resultados.

- 5.- Se ha presentado una nueva transformación del problema de colocación en asignación cuadrática, basada en la introducción de un factor de peso que depende del número de elementos de cada conexión, ajustando su expresión con el fin de equilibrar la influencia de las distribuciones más frecuentes de los elementos de una conexión.
- 6.- Se ha evaluado el comportamiento de la transformación propuesta frente a la transformación clásica, basada en utilizar un factor de peso unidad, y frente a la resolución directa del problema de colocación por intercambio exhaustivo por parejas, con los datos procedentes de los ejemplos de diseño de tarjetas de circuitos impresos.

Este estudio pone de manifiesto la superioridad de la nueva transformación, frente a la clásica y al tra -

tamiento directo, tanto en el tiempo de ejecución como en el valor final alcanzado por la función de evaluación.

- 7.- Se consigue proporcionar a la función de evaluación una información sobre la disponibilidad de pistas, con el fin de evitar congestiones en el trazado de conexiones, a través de una definición de distancia que da más peso a la dirección transversal a las pastillas de circuitos integrados, que a la dirección longitudinal.
- 8.- El algoritmo de colocación, en su forma definitiva, consiste en la transformación del problema de colocación en otro de asignación cuadrática, usando la nueva transformación propuesta, la resolución del problema de asignación cuadrática utilizando el nuevo algoritmo y la utilización de la nueva distancia para el cálculo de la función de evaluación.
- 9.- Se ha utilizado, el algoritmo de conexionado del que disponemos, como instrumento para la medida definitiva de la eficacia de una colocación. Para cada una de las tarjetas estudiadas se analiza la eficacia de una colocación manual, proporcionada por un diseñador experto, de una colocación obtenida y por la resolución directa del problema de colocación y de la colocación proporcionada por el algoritmo que presentamos.

Este análisis pone de manifiesto la mayor eficacia del nuevo algoritmo, como puede verse en la tabla nº 10.

- 10.- Como aplicación práctica se han integrado los programas de colocación en un sistema de diseño, que incluye colocación y trazado automático de conexiones, usando el mismo lenguaje orientado al problema para la entrada de datos.

Los programas de colocación pueden trabajar con problemas que tengan hasta 150 elementos y 300 conexiones, utilizando un miniordenador con 32K-palabras.

A P E N D I C E I

DESCRIPCION DE LAS TARJETAS UTILIZADAS

En este trabajo se han utilizado cuatro tarjetas de circuitos impresos para el estudio del problema de colocación. Las tres primeras tarjetas, han sido seleccionadas entre las - que distintos usuarios del sistema de trazado automático de co nexiones, han traído al Instituto de Automática Industrial para su resolución. Han sido elegidas tratando de que su tamaño cubra la gama a estudiar y que sus componentes sean fundamenta lemente circuitos integrados.

A continuación se resumen las características de es tas tarjetas:

Tarjeta nº 1: UCA. Es una tarjeta pequeña, diseñada en el Insti tuto de Automática Industrial, que con-- tiene 15 circuitos integrados, dos conecto res para entrada salida, cinco resistencias y dos condensadores. El motivo de su elec ción es disponer de un ejemplo sencillo y de pequeño tamaño.

Tarjeta nº 2: IKER3. Es una tarjeta que contiene 49 circuitos integrados y dos conectores para entrada salida.

Tarjeta nº 3: FLIFLO. Esta tarjeta contiene 56 circuitos integrados, cuatro conectores para entrada - salida y un condensador.

Tarjeta nº 4: IL1. Es una tarjeta que ha sido tomada del trabajo (10). Consta de 101 circuitos integrados y 30 conectores para entrada salida. Esta tarjeta procede de la unidad de control del ordenador ILLIAC IV.

A continuación, se presentan en el lenguaje orientado al problema (2), la descripción completa de las tres primeras tarjetas utilizadas.

```

UCAOR. TAR
/
TA=65*39.5; MA=2; ES=1;
EL=C16, C18, C15, C13, C14, C12, C110, C111, C112, C146; DL14,
    C17, C11, C114, C113, C19; DL16;
PO=C16(13, 34), C17(26, 34), C18(42, 34), C15(55, 34),
    C13(55, 24), C14(42, 24), C12(27, 24), C11(12, 24),
    C114(12, 14), C113(26, 14), C110(42, 14), C111(55, 14),
    C112(55, 4), C19(41, 4), C146(13, 4);
EL=R1, R2, R3, R4, R5, C2; DP5, C1; DP2;
QR=S;
PO=C1(37, 3), C2(35, 3), R1(32, 3), R2(30, 3), R3(28, 3), R4(26, 3), R5(24, 3);
EL=ESA, ESB; CT27;
PO=ESA(1, 6, 5), ESB(1, 5, 7);
LP2=MASA: (1, 34, 5)(20, 34, 5), (19, 5, 35)(19, 5, 1), (19, 5, 1)(63, 1),
    (62, 5, 1)(62, 5, 34), (48, 5, 1)(48, 5, 34), (33, 5, 1)(33, 5, 34);
LP2=A15V: (11, 6)(54, 5, 6), (54, 6)(54, 37), (40, 6)(40, 37), (25, 6)(25, 37),
    (11, 6)(11, 37);
CO=3, C11, 21, ESA, (1, 12), C12, 1, C13, 3, C15;
CO=(2, 3), C146, 1, R5;
CO=1, C146, 7, ESA;
CO=3, C114, 8, ESA;
CO=5, C114, 7, ESB;
CO=7, C114, 8, ESB;
CO=10, C114, 10, ESA;
CO=12, C114, 10, ESB;
CO=15, C114, 4, ESB;
CO=23, ESA, 1, (C113, C114), 14, C146;
CO=9, C113, 15, ESA;
CO=10, C113, 15, ESB;
CO=12, C113, 13, ESB;
CO=15, C113, 13, ESA;
CO=6, C11, 5, C16, 8, C15, 4, C114, 3, C13;
CO=15, C11, 2, C13, (1, 11), C15;
CO=14, C11, 21, ESB, 13, C12, 4, C15, 12, C14, (11, 3), C16;
CO=12, C11, 9, C18;
CO=11, C11, 9, C16, 2, C18;
CO=20, ESA, 13, C16, 7, C11, 8, C13, 11, C14, (13, 5), C15;
CO=5, C11, 20, ESB, (12, 4), C16, 12, C15, 2, C12;
CO=4, C11, 5, C13, 9, C14, 2, C15;
CO=2, C11, 10, C14, 4, C13;
CO=(8, 11), C12;
CO=10, C12, 14, C114;
CO=9, C12, 10, C16;
CO=6, (C12, C16);
CO=5, C12, 13, C14;
CO=4, C12, 11, C114;
CO=3, C12, 11, C13;
CO=13, C13, 6, C13, 2, C14;
CO=(12, 9), C13;
CO=10, C13, 9, C114;
CO=5, C14, 6, C15;
CO=4, C14, 9, C15;
CO=3, C14, 10, C15;
CO=1, C14, 6, C114;
CO=18, ESA, (8, 9), C112, 8, (C110, C16);
CO=2, C16, 17, ESA, 12, C111;
CO=1, C16, 18, ESB, 13, C111;
CO=15, C17, 5, C18;
CO=14, C17, 8, C18, 14, C113, 11, C113;
CO=13, C17, 3, C18;

```

CO=(12, 4). CI7, 11. CI10;
 CO=(5, 6, 10, 11). CI7, 12. CI10, 1. R1;
 CO=3. CI7, 6. CI8, 12. CI12, 2. CI10, 6. CI9;
 CO=2. CI7, 1. CI10;
 CO=1. CI7, 13. CI12;
 CO=(13, 10). CI8;
 CO=(12, 4). CI8;
 CO=11. CI8, 11. CI9;
 CO=1. (CI8, CI12), 3. CI10;
 CO=15. CI9, 1. C1;
 CO=14. CI9, 2. C1, 1. R2;
 CO=(13, 3). CI9, 13. CI10, 17. ESB;
 CO=(10, 12). CI9, 2. CI14;
 CO=(7, 5). CI9;
 CO=4. (CI9, CI46), 1. R4;
 CO=2. CI9, 1. R3, 1. C2;
 CO=1. CI9, 2. C2;
 CO=(10, 5). CI10;
 CO=9. CI10, (8, 9, 11). CI11;
 CO=6. CI10, 4. CI11;
 CO=4. CI10, 5. ESB;
 CO=(10, 5). CI11;
 CO=(6, 3). CI11;
 CO=(2, 1). CI11, 11. CI12;
 CO=(10, 6). CI12;
 CO=(5, 3). CI12, 11. ESB;
 CO=(4, 2). CI12, 11. ESA;
 CO=A15V, 26. ESA, 1. CI1, 14. (CI2, CI3, CI5, CI6, CI8, CI12, CI4, CI11,
 CI10), 16. (CI14, CI7, CI9, CI13), 2. (R1, R2, R3, R4, R5);
 CO=MASA, 27. ESA, (8, 9). (CI1, CI7), 7. (CI46, CI3, CI2, CI5, CI4, CI10,
 CI11, CI7, CI6, CI8, CI12), 8. (CI9, CI13, CI14);
 EJ=;

IKER3.TAR (PLACA CMOS)

TA=100*55; MA=2; ES=1;

EL=U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13,
 U14, U15, U16, U17, U18, U19, U20, U21, U22, U23, U24,
 U25, U26, U27, U28, U29, U30, U31, U32, U33, U34, U35,
 U36, U37, U38, U39, U40, U41, U42, U43, U44, U45, U49; DL16,
 U46, U47, U48; DL14;

EL=D, I; CT49;

OR=E;

PO=U1(22, 53), U2(22, 45), U3(34, 53), U4(34, 45), U5(46, 53),
 U6(46, 45), U7(58, 53), U8(58, 45), U9(70, 53), U10(70, 45),
 U11(82, 53), U12(82, 45), U13(82, 37), U14(94, 37),
 U15(82, 29), U16(94, 29), U17(82, 21), U18(94, 21),
 U19(22, 13), U20(22, 5), U21(34, 13), U22(34, 5), U23(46, 13),
 U24(46, 5), U25(58, 13), U26(58, 5), U27(70, 13), U28(70, 5),
 U29(82, 13), U30(82, 5), U31(94, 13), U32(94, 5), U33(22, 38),
 U34(22, 32), U35(22, 26), U36(22, 20), U37(34, 38),
 U38(34, 32), U39(34, 26), U40(34, 20), U41(46, 38),
 U42(46, 32), U43(46, 26), U44(46, 20), U45(58, 38),
 U46(58, 20), U47(70, 38), U48(70, 20), U49(94, 53);

OR=N;

PO=I(4, 52), D(4, 51, 5);

CO=40, D; 12, U34;

CO=40, I; 9, U34;

CO=39, I; 7, U34;

CO=39, D; 4, U34;

CO=38, D; 12, U33;

CO=38, I; 9, U33;

CO=37, I; 4, U33;

CO=37, D; 7, U33;

CO=14, U34, 3, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13, U14, U15, U16, U17,
 U18, U19, U20, U21, U22, U23, U24, U25, U26, U27, U28, U29, U30, U31, U32);

CO=11, U34, 2, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13,
 U14, U15, U16, U17, U18, U19, U20, U21, U22, U23, U24, U25, U26, U27, U28,
 U29, U30, U31, U32);

CO=5, U34, 1, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13, U14,
 U15, U16, U17, U18, U19, U20, U21, U22, U23, U24, U25, U26, U27,
 U28, U29, U30, U31, U32);

CO=14, U33, 16, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13,
 U14, U15, U16, U17, U18, U19, U20, U21, U22, U23, U24, U25, U26, U27,
 U28, U29, U30, U31, U32);

CO=11, U33, 4, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13,
 U14, U15, U16, U17, U18, U19, U20, U21, U22, U23, U24, U25, U26,
 U27, U28, U29, U30, U31, U32);

CO=2, U33, 5, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12,
 U13, U14, U15, U16, U17, U18, U19, U20, U21, U22,
 U23, U24, U25, U26, U27, U28, U29, U30, U31, U32);

CO=5, U33, 6, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12,
 U13, U14, U15, U16, U17, U18, U19, U20, U21, U22, U23,
 U24, U25, U26, U27, U28, U29, U30, U31, U32);

CO=30, I; 13, U35;

CO=32, I; 10, U35;

CO=30, D; 3, U35;

CO=32, D; 6, U35;

CO=29, I; 13, U36;

CO=31, I; 10, U36;

CO=29, D; 3, U36;

CO=31, D; 6, U36;

CO=(5, 7), U36, 10, (U1, U3, U5, U7, U9, U11, U13, U15, U17,
 U19, U21, U23, U25, U27, U29, U31);

CO=(4, 2), U36, 11, (U1, U3, U5, U7, U9, U11, U13, U15, U17,
U19, U21, U23, U25, U27, U29, U31);

CO=(9, 11), U36, 12, (U1, U3, U5, U7, U9, U11, U13, U15, U17,
U19, U21, U23, U25, U27, U29, U31);

CO=(12, 14), U36, 13, (U1, U3, U5, U7, U9, U11, U13, U15, U17,
U19, U21, U23, U25, U27, U29, U31);

CO=(5, 7), U35, 10, (U2, U4, U6, U8, U10, U12, U14, U16, U18,
U20, U22, U24, U26, U28, U30, U32);

CO=(2, 4), U35, 11, (U2, U4, U6, U8, U10, U12, U14, U16, U18,
U20, U22, U24, U26, U28, U30, U32);

CO=(9, 11), U35, 12, (U2, U4, U6, U8, U10, U12, U14, U16, U18,
U20, U22, U24, U26, U28, U30, U32);

CO=(12, 14), U35, 13, (U2, U4, U6, U8, U10, U12, U14, U16, U18,
U20, U22, U24, U26, U28, U30, U32);

CO=8, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13, U14, U15, U16,
U17, U18, U19, U20, U21, U22, U23, U24, U25, U26, U27, U28, U29,
U30, U31, U32), 10, U40;

CO=15, (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13, U14, U15, U16, U17,
U18, U19, U20, U21, U22, U23, U24, U25, U26, U27, U28, U29, U30, U31, U32),
7, U40, 2, U46, 6, D;

CO=(9, 14), (U1, U2), 3, U44;

CO=(9, 14), (U3, U4), 5, U44;

CO=(9, 14), (U5, U6), 7, U44;

CO=(9, 14), (U7, U8), 9, U44;

CO=(9, 14), (U9, U10), 11, U44;

CO=(9, 14), (U11, U12), 13, U44;

CO=(9, 14), (U13, U14), 3, U43;

CO=(9, 14), (U15, U16), 5, U43;

CO=(9, 14), (U17, U18), 7, U43;

CO=(9, 14), (U19, U20), 9, U43;

CO=(9, 14), (U21, U22), 3, U42;

CO=(9, 14), (U23, U24), 5, U42;

CO=(9, 14), (U25, U26), 7, U42;

CO=(9, 14), (U27, U28), 9, U42;

CO=(9, 14), (U29, U30), 11, U42;

CO=(9, 14), (U31, U32), 13, U42;

CO=36, D, 12, U37;

CO=36, I, 9, U37;

CO=35, I, 7, U37;

CO=35, D, 4, U37, 6, U39;

CO=14, U37, 1, U38, 1, U39;

CO=11, U37, 2, U38, 2, U39;

CO=5, U37, 3, U38, 3, U39;

CO=3, U37, 6, U38;

CO=5, U38, 5, U39, 6, U48, 12, U40;

CO=4, U38, 4, U39, 3, U40;

CO=15, U38, 2, U44;

CO=14, U38, 4, U44;

CO=13, U38, 6, U44;

CO=12, U38, 10, U44;

CO=11, U38, 12, U44;

CO=10, U38, 14, U44;

CO=9, U38, 2, U43;

CO=7, U38, 4, U43;

CO=15, U39, 6, U43;

CO=14, U39, 10, U43;

CO=13, U39, 2, U42;

CO=12, U39, 4, U42;

CO=11, U39, 6, U42;

CO=10, U39, 10, U42;

CO=9, U39, 12, U42;

CO=7, U39, 14, U42;

CO=(1, 15), (U42, U43, U44), 1, U45;

CO=34, D, 9, U41;

CO=34, I, 12, U41;

CO=33, I, 4, U41;

CO=33, D, 7, U41;
CO=11, U41, 1, U49;
CO=10, U41, 2, U49;
CO=14, U41, 4, U49;
CO=13, U41, 3, U49;
CO=2, U41, 12, U49;
CO=3, U41, 11, U49;
CO=5, U41, 9, U49;
CO=6, U41, 10, U49;
CO=15, U49, 1, U48;
CO=14, U49, 2, U48;
CO=6, U49, 4, U48;
CO=7, U49, 5, U48;
CO=(1, 15), (U35, U36), 8, U48;
CO=6, 1, 4, U40;
CO=8, 1, 12, U48, 9, U40, 1, U46;
CO=13, U40, 13, U48;
CO=6, U40, (10, 9), U48;
CO=5, 1, 2, U47, 5, U45;
CO=20, 1, 2, U45;
CO=1, U47, 6, U45;
CO=3, U46, 3, U45;
EJ=;

FLIFLO. TAR

TA=93*77; MA=2; ES=1;

EL=N1, N2, N3, N4, N5, N6, N7, N8, N9, N10, N11, N12,
 N13, N14, N15, N16, N17, N18, N19, N20, N21, N22,
 N23, N24, N25, N26, N27, N28, N29, N30, N31, N32, N33, N34,
 N35, N36, N37, N38, N39, N40, N41, N42, N43, N44,
 N45, N46, N47, N48, N49, N50, N51, N52, N53, N54,
 N55, N56; DL16;

OR=E;

PO=N1(79, 5), N2(79, 15), N3(79, 25), N4(79, 35), N5(79, 45), N6(79, 55),
 N7(79, 65), N8(79, 75),
 N9(67, 5), N10(67, 15), N11(67, 25), N12(67, 35),
 N13(67, 45), N14(67, 55), N15(67, 65), N16(67, 75),
 N17(55, 5), N18(55, 15), N19(55, 25), N20(55, 35),
 N21(55, 45), N22(55, 55), N23(55, 65), N24(55, 75),
 N25(43, 5), N26(43, 15), N27(43, 25), N28(43, 35),
 N29(43, 45), N30(43, 55), N31(43, 65), N32(43, 75),
 N33(31, 5), N34(31, 15), N35(31, 25), N36(31, 35);
 PO=N37(31, 45), N38(31, 55), N39(31, 65), N40(31, 75),
 N41(19, 5), N42(19, 15), N43(19, 25), N44(19, 35),
 N45(19, 45), N46(19, 55), N47(19, 65), N48(19, 75),
 N49(8, 5), N50(8, 15), N51(8, 25), N52(8, 35), N53(8, 45),
 N54(8, 55), N55(8, 65), N56(8, 75);

EL=CONA, CONB, CONC, COND: CT33;

OR=N;

PO=CONA(92, 76), CONB(92, 33), CONC(91, 76, 5), COND(91, 32, 5);

OR=E;

EL=C2: DP9;

PO=C2(67, 58, 5)N;

LP2=ASV: (8, 5, 5)(8, 5, 76, 5), (20, 5, 5)(20, 5, 76, 5), (32, 5, 5)
 (32, 5, 76, 5), (44, 5, 5)(44, 5, 76, 5), (56, 5, 5)(56, 5, 76, 5),
 (68, 5, 5)(68, 5, 76, 5), (80, 5, 5)(80, 5, 76, 5);

LP3=MASA: (9, 5, 5)(9, 5, 76, 5), (21, 5, 5)(21, 5, 76, 5), (33, 5, 5)
 (33, 5, 76, 5), (45, 5, 5)(45, 5, 76, 5), (57, 5, 5)(57, 5, 76, 5),
 (69, 5, 5)(69, 5, 76, 5), (81, 5, 5)(81, 5, 76, 5), (0, 5, 18)(82, 5, 18),
 (0, 5, 38)(82, 5, 38), (0, 5, 58)(82, 5, 58);

LI=STOP: (86, 5, 58)(87, 5, 58), (87, 5, 58)(87, 5, 59), (87, 5, 59)
 (86, 5, 59), (86, 5, 59)(86, 5, 58), (86, 5, 49)(87, 5, 49),
 (87, 5, 49)(87, 5, 50), (87, 5, 50)(86, 5, 50), (86, 5, 50)
 (86, 5, 49);

CO=ASV, 16. (N1, N2, N3, N4, N5, N6, N7, N8, N9, N10, N11, N12, N13,
 N14, N15, N16, N17, N18, N19, N20, N21, N22, N23, N24,
 N25, N26, N27, N28, N29, N30, N31, N32, N33, N34, N35,
 N36, N37, N38, N39, N40, N41, N42, N43, N44, N45, N46,
 N47, N48, N49, N50, N51, N52, N53, N54, N55, N56);

CO=MASA, 8. (N1, N2, N3, N4, N5, N6, N7, N8, N9, N10, N11, N12, N13, N14,
 N15, N16, N17, N18, N19, N20, N21, N22, N23, N24, N25, N26,
 N27, N28, N29, N30, N31, N32, N33, N34, N35, N36, N37, N38, N39,
 N40, N41, N42, N43, N44, N45, N46, N47, N48);

CO=6, N5, 30, CONC;

CO=7, N6, 21, CONA;

CO=1, C2, 19, (CONA, CONC);

CO=2, C2, 28, (CONA, CONC);

CO=2, N9, 26, COND;

CO=4, N9, 26, CONB;

CO=12, N9, 30, COND;

CO=10, N9, 30, CONB;

CO=6, N9, 25, COND;

CO=6, N10, 17, CONB;

CO=2, N10, 17, COND;

CO=14. N9, 31. CONB;
 CO=10. N53, 24. CONA;
 CO=12. N53, 31. CONC;
 CO=14. N53, 30. CONA;
 CO=4. N46, 20. CONC;
 CO=2. N16, 4. CONA;
 CO=3. N16, 8. CONC;
 CO=5. N16, 8. CONA;
 CO=6. N16, 16. CONC;
 CO=7. N16, 7. CONA;
 CO=3. N42, 24. COND;
 CO=5. N24, 4. CONC;
 CO=5. N23, 9. CONA;
 CO=5. N22, 20. CONA;
 CO=5. N21, 24. CONC;
 CO=5. N20, 33. CONA;
 CO=5. N19, 4. COND;
 CO=5. N18, 13. CONB;
 CO=5. N17, 25. CONB;
 CO=3. N8, 6. CONC;
 CO=4. N8, 6. CONA;
 CO=5. N8, 7. CONC;
 CO=6. N8, 5. CONA;
 CO=7. N8, 5. CONC;
 CO=3. N11, 7. CONB;
 CO=4. N11, 6. COND;
 CO=5. N11, 6. CONB;
 CO=6. N11, 5. COND;
 CO=7. N11, 5. CONB;
 CO=3. N3, 9. COND;
 CO=4. N3, 9. CONB;
 CO=5. N3, 8. COND;
 CO=6. N3, 8. CONB;
 CO=7. N3, 7. COND;
 CO=3. N7, 13. CONC;
 CO=4. N7, 13. CONA;
 CO=5. N7, 14. CONC;
 CO=6. N7, 14. CONA;
 CO=7. N7, 15. CONC;
 CO=3. N12, 1. COND;
 CO=4. N12, 33. CONC;
 CO=5. N12, 32. CONA;
 CO=6. N12, 32. CONC;
 CO=7. N12, 31. CONA;
 CO=3. N4, 2. COND;
 CO=4. N4, 3. CONB;
 CO=5. N4, 2. CONB;
 CO=6. N4, 3. COND;
 CO=7. N4, 4. CONB;
 CO=3. N15, 12. CONC;
 CO=4. N15, 11. CONA;
 CO=5. N15, 11. CONC;
 CO=6. N15, 10. CONA;
 CO=7. N15, 10. CONC;
 CO=3. N1, 28. COND;
 CO=4. N1, 28. CONB;
 CO=5. N1, 27. COND;
 CO=6. N1, 27. CONB;
 CO=7. N1, 29. COND;
 CO=3. N14, 21. CONC;
 CO=4. N14, 18. CONC;
 CO=5. N14, 17. CONA;
 CO=6. N14, 16. CONA;
 CO=7. N14, 17. CONC;
 CO=3. N2, 18. COND;
 CO=4. N2, 19. CONB;

135

CO=5. N2, 20. CONB;
 CO=6. N2, 20. COND;
 CO=7. N2, 21. CONB;
 CO=3. N13, 27. CONC;
 CO=4. N13, 26. CONA;
 CO=5. N13, 26. CONC;
 CO=6. N13, 25. CONA;
 CO=7. N13, 25. CONC;
 CO=3. N6, 22. CONA;
 CO=4. N6, 23. CONC;
 CO=5. N6, 23. CONA;
 CO=6. N6, 22. CONC;
 CO=3. N5, 27. CONA;
 CO=4. N5, 29. CONC;
 CO=5. N5, 29. CONA;
 CO=3. N9, 4. N25, 4. N26, 4. N27, 4. N28, 4. N29, 4. N30, 4. N31, 4. N32;
 CO=5. N9, 5. N25, 5. N26, 5. N27, 5. N28, 5. N29, 5. N30, 5. N31, 5. N32;
 CO=11. N9, 12. N25, 12. N26, 12. N27, 12. N28, 12. N29, 12. N30, 12. N31, 12. N32;
 CO=9. N9, 13. N25, 13. N26, 13. N27, 13. N28, 13. N29, 13. N30, 13. N31, 13. N32;
 CO=7. N9, 4. N33, 4. N34, 4. N35, 4. N36, 4. N37, 4. N38, 4. N39, 4. N40;
 CO=7. N10, 5. N33, 5. N34, 5. N35, 5. N36, 5. N37, 5. N38, 5. N39, 5. N40;
 CO=3. N10, 12. N33, 12. N34, 12. N35, 12. N36, 12. N37, 12. N38, 12. N39, 12. N40;
 CO=13. N9, 13. N33, 13. N34, 13. N35, 13. N36, 13. N37, 13. N38, 13. N39, 13. N40;
 CO=9. N53, 6. N53; FUERA;
 CO=11. N53, 4. N53;
 CO=13. N53, 2. N53;
 CO=7. N53, 1. N54, 11. N24, 11. N23, 11. N22, 11. N21, 11. N20, 11. N19, 11. N18,
 11. N17;
 CO=5. N53, 2. N45, 10. N24, 10. N23, 10. N22, 10. N21, 10. N20, 10. N19, 10. N18,
 10. N17;
 CO=3. N53, 3. N45, 9. N24, 9. N23, 9. N22, 9. N21, 9. N20, 9. N19, 9. N18, 9. N17;
 CO=5. N46, 5. N45;
 CO=4. N45, 9. N16, 4. N10;
 CO=5. N10, 2. N42;
 CO=12. N42, (5, 6, 10, 11). N42;
 CO=4. N42, 7. N24, 7. N23, 7. N22, 7. N21, 7. N20, 7. N19, 7. N18, 7. N17;
 CO=7. N42, 9. N42, 1. N25, 1. N26, 1. N27, 1. N28, 1. N29, 1. N30, 1. N31,
 1. N32, 1. N33, 1. N34, 1. N35, 1. N36, 1. N37, 1. N38, 1. N39, 1. N40;
 CO=15. N46, 14. N10;
 CO=14. N45, 10. N10;
 CO=13. N45, 12. N10;
 CO=12. N45, 14. N46;
 CO=11. N45, 12. N46;
 CO=10. N45, 10. N46;
 CO=9. N45, 2. N46;
 CO=7. N45, 6. N46;
 CO=13. N10, 9. N25, 9. N33;
 CO=9. N10, 9. N26, 9. N34;
 CO=11. N10, 9. N27, 9. N35;
 CO=13. N46, 9. N28, 9. N36;
 CO=11. N46, 9. N29, 9. N37;
 CO=9. N46, 9. N30, 9. N38;
 CO=3. N46, 9. N31, 9. N39;
 CO=7. N46, 9. N32, 9. N40;
 CO=2. N25, 4. N24, 14. N8;
 CO=7. N25, 4. N23, 13. N8;
 CO=10. N25, 4. N22, 12. N8;
 CO=15. N25, 4. N21, 11. N8;
 CO=2. N33, 4. N20, 10. N8;
 CO=7. N33, 4. N19, 14. N11;
 CO=10. N33, 4. N18, 13. N11;
 CO=15. N33, 4. N17, 12. N11;
 CO=2. N26, 3. N24, 11. N11;
 CO=7. N26, 3. N23, 10. N11;
 CO=10. N26, 3. N22, 14. N3;
 CO=15. N26, 3. N21, 13. N3;

CO=2. N34, 3. N20, 12. N3;
 CO=7. N34, 3. N19, 11. N3;
 CO=10. N34, 3. N18, 10. N3;
 CO=15. N34, 3. N17, 14. N7;
 CO=2. N27, 2. N24, 13. N7;
 CO=7. N27, 2. N23, 12. N7;
 CO=10. N27, 2. N22, 11. N7;
 CO=15. N27, 2. N21, 10. N7;
 CO=2. N35, 2. N20, 14. N12;
 CO=7. N35, 2. N19, 13. N12;
 CO=10. N35, 2. N18, 12. N12;
 CO=15. N35, 2. N17, 11. N12;
 CO=2. N28, 1. N24, 10. N12;
 CO=7. N28, 1. N23, 14. N4;
 CO=10. N28, 1. N22, 13. N4;
 CO=15. N28, 1. N21, 12. N4;
 CO=2. N36, 1. N20, 11. N4;
 CO=7. N36, 1. N19, 10. N4;
 CO=10. N36, 1. N18, 14. N15;
 CO=15. N36, 1. N17, 13. N15;
 CO=2. N29, 15. N24, 12. N15;
 CO=7. N29, 15. N23, 11. N15;
 CO=10. N29, 15. N22, 10. N15;
 CO=15. N29, 15. N21, 14. N1;
 CO=2. N37, 15. N20, 13. N1;
 CO=7. N37, 15. N19, 12. N1;
 CO=10. N37, 15. N18, 11. N1;
 CO=15. N37, 15. N17, 10. N1;
 CO=2. N30, 14. N24, 14. N14;
 CO=7. N30, 14. N23, 13. N14;
 CO=10. N30, 14. N22, 12. N14;
 CO=15. N30, 14. N21, 11. N14;
 CO=2. N38, 14. N20, 10. N14;
 CO=7. N38, 14. N19, 14. N2;
 CO=10. N38, 14. N18, 13. N2;
 CO=15. N38, 14. N17, 12. N2;
 CO=2. N31, 13. N24, 11. N2;
 CO=7. N31, 13. N23, 10. N2;
 CO=10. N31, 13. N22, 14. N13;
 CO=15. N31, 13. N21, 13. N13;
 CO=2. N39, 13. N20, 12. N13;
 CO=7. N39, 13. N19, 11. N13;
 CO=10. N39, 13. N18, 10. N13;
 CO=15. N39, 13. N17, 14. N6;
 CO=2. N32, 12. N24, 13. N6;
 CO=7. N32, 12. N23, 12. N6;
 CO=10. N32, 12. N22, 11. N6;
 CO=15. N32, 12. N21, 10. N6;
 CO=2. N40, 12. N20, 14. N5;
 CO=7. N40, 12. N19, 13. N5;
 CO=10. N40, 12. N18, 12. N5;
 CO=15. N40, 12. N17, 11. N5;
 EU=;

A P E N D I C E I I .

Cada una de las tarjetas presentadas en el apéndice I, da lugar a un problema de colocación.

Presentamos a continuación, a modo de ejemplo, la descripción de las conexiones correspondientes a la tarjeta nº 2.

DK1: IKER3 ST

CONEXION: 1

51 34

CONEXION: 2

51 34

CONEXION: 3

51 34

CONEXION: 4

51 34

CONEXION: 5

51 33

CONEXION: 6

51 33

CONEXION: 7

51 33

CONEXION: 8

51 33

CONEXION: 9

34	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32							

CONEXION: 10

34	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32							

CONEXION: 11

34	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32							

CONEXION: 12

33	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32							

CONEXION: 13

33	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32							

CONEXION: 14

33	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32							

CONEXION: 15

33	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32							

CONEXION: 16

52 35

CONEXION: 17

52 35

CONEXION: 18

52 35

CONEXION: 19

52 35

CONEXION: 20

52 36

CONEXION: 21

52 36

CONEXION: 22

52 36

CONEXION: 23

52 36

CONEXION: 24

36	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31			
----	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	--	--	--

CONEXION: 25

36	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31			
----	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	--	--	--

CONEXION: 26

36	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31			
----	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	--	--	--

CONEXION: 27

36	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31			
----	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	--	--	--

139

CONEXION: 28
35 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32

CONEXION: 29
35 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32

CONEXION: 30
35 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32

CONEXION: 31
35 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32

CONEXION: 32
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 40

CONEXION: 33
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 40 47 56

CONEXION: 34
1 2 44

CONEXION: 35
3 4 44

CONEXION: 36
5 6 44

CONEXION: 37
7 8 44

CONEXION: 38
9 10 44

CONEXION: 39
11 12 44

CONEXION: 40
13 14 43

CONEXION: 41
15 16 43

CONEXION: 42
17 18 43

CONEXION: 43
19 20 43

CONEXION: 44
21 22 42

CONEXION: 45
23 24 42

CONEXION: 46
25 26 42

CONEXION: 47
27 28 42

CONEXION: 48
29 30 42

CONEXION: 49
31 32 42

CONEXION: 50
51 37

CONEXION: 51
51 37

CONEXION: 52
52 37

CONEXION: 53
52 37 39

CONEXION: 54
37 38 39

CONEXION: 55
37 38 39

CONEXION: 56
37 38 39

CONEXION: 57
37 38

CONEXION: 58
38 39 40 40

CONEXION: 59
38 39 40

CONEXION: 60
38 44
CONEXION: 61
38 44
CONEXION: 62
38 44
CONEXION: 63
38 44
CONEXION: 64
38 44
CONEXION: 65
38 44
CONEXION: 66
38 43
CONEXION: 67
38 43
CONEXION: 68
39 43
CONEXION: 69
39 43
CONEXION: 70
39 42
CONEXION: 71
39 42
CONEXION: 72
39 42
CONEXION: 73
39 42
CONEXION: 74
39 42
CONEXION: 75
39 42
CONEXION: 76
42 43 44 45
CONEXION: 77
52 41
CONEXION: 78
52 41
CONEXION: 79
52 41
CONEXION: 80
52 41
CONEXION: 81
41 46
CONEXION: 82
41 46
CONEXION: 83
41 46
CONEXION: 84
41 46
CONEXION: 85
41 46
CONEXION: 86
41 46
CONEXION: 87
41 46
CONEXION: 88
41 46
CONEXION: 89
46 49
CONEXION: 90
46 49
CONEXION: 91
46 49
CONEXION: 92
46 49

1110

141

CONEXION: 93
35 36 49
CONEXION: 94
56 40
CONEXION: 95
55 49 40 47
CONEXION: 96
40 49
CONEXION: 97
40 49
CONEXION: 98
56 48 45
CONEXION: 99
54 45
CONEXION: 100
54 45
CONEXION: 101
48 45
CONEXION: 102
47 45

B I B L I O G R A F I A

Las referencias que se presentan guardan una estricta relación con los temas tratados en esta memoria. No hemos hecho referencia a otras obras de carácter general que ha sido necesario consultar.

Estas referencias están agrupadas en dos partes, en la primera se presentan ordenadamente las citadas explícitamente en el texto, en la segunda aparecen otras publicaciones específicas consultadas.

Referencias que se citan en el texto

1. W. G. MAGNUSON. "Computer aided design and design automation in Europe". SIGDA Newsletter - A.C.M., nº 1, vol. 6, pp. 3-25, March 1976.
2. M^a. T. DE PEDRO LUCIO. "Sistema de inteligencia artificial aplicado al diseño automático. Lenguaje orientado al problema". Tesis Doctoral, Universidad Complutense de Madrid, Facultad de Ciencias, Junio 1977.
3. K. H. KHOKHANI; A.M. PATEL. "The chip layout problem: A - placement procedure for LSI". Proc. of the 14th Design - Automation Conference. pp. 291-302., New Orleans, June -- 1977.
4. J.L. BENTLEY; J.H. FRIEDMAN. "Fast algorithms for constructing minimal spanning trees in coordinate spaces". IEEE -- Trans. on Comp., nº 2, vol. C-27, pp. 97-105, febrero 1978.
5. M. HANAN; P.K. WOLFF; B.J. AGULE. "A study of placement -- techniques". Journal of Design Automation and Fault Tolerant Computing, nº 1, vol. 1, pp. 28-61, octubre 1976.
6. M.A. BREUER. "Design automation of digital systems: Theory and Techniques". Prentice-Hall, New York, 1972.

7. M. HANAN; J.M. KURTZBERG. "A review of the placement and - quadratic assignment problems". SIAM Rev., vol. 14, n° 2, pp. 324-341, april 1972.
8. K.H. HOSKING. "The use of the computer program suite orchard for designing the artwork for double-sided printed boards". The Marconi Review, vol. 37, n° 194, thied quarter 1974.
9. W.G. MAGNUSON. "A comparison of constructive placement algorithms. IEEE, 1977 Regim 6 Conference Record ' Electronics Serving Mankind', pp. 28-32, may 1977.
10. J.E. STEVENS. "Fast heuristic techniques for placing and - wiring printed circuit boards". Ph. D. Thesis, Comp. Sci. Dept. Univ. of Illinois. 1972.
11. L. Steimberg. "The backboard wiring problem: a placement - algorithm". SIAM Rev., vol. 3, n° 1, pp. 37-50, Jan. 1961.
12. T.C. KOOPMANS; M.J. BECKMANN. "Assignment problems and the location of economic activities". Econometrica. vol. 25, - n° 1, pp. 53-76, 1957.
13. P.C. GILMORE. "Optimal and suboptimal algorithms for the - quadratic assignment problem". J. Soc. Indust. Appl. Math., vol. 10, n° 2, pp. 305-313, June 1962.
14. E.E. LAWLER. "The quadratic assignment problem". Management Science, vol. 9, pp. 586-599, 1963.

15. R.A. RUTMAN. "An algorithm for placement of interconnected elements based on minimum wire length". Proc. 1964 SJCC, - pp. 477-491.
16. H.W. KUHN. "Hungarian method for the assignment problem". Naval Res. Logist. Quart., 2, pp. 88-97, 1955.
17. J. MUNKRES. "Algorithms for the assignment and transportation problems". J. Soc. Indust. Appl. Math., 5, pp. 32-38, 1957.
18. R.G. GARSIDE; T.A. NICHOLSON. "Permutation procedure for - the backboard wiring problem". Proc. IEE, 115, pp. 27-30, Jan. 1968.
19. C.J. FISK; D.L. CASKEY; L.E. WEST. "ACCEL: Automated circuit card etching layout". Proc. IEEE, vol. 55, n° 11, -- pp. 1971-1982, nov. 1967.
20. A.M. PATEL; C.G. DE WALD; L.C. COTE. "Prepartitioning as a means of enhancing pairwise interchange solutions to quadratic assignment problems". Comp. and Ops. Res., vol. 3, pp. 49-56, 1976.
21. R. BORST; C.G. DE WALD. "A solution technique for placing digital integrated circuitry on printed circuit boards". Comp. and Ops. Res., vol. 1, pp. 161-180, 1974.

22. S.C. HOFFMAN. "Automatic gate allocation placement and routing". Proc. of the Symposium on Computer Aided Design of - Digital Electronic Circuits and Systems. pp. 299-306, Brussels 27-29, Nov. 1978.
23. R.H. GLASER. "A quasi-simplex method for designing suboptimal packages for electronic building blocks". Proc. 1959 - Comp. Appl. Symp., pp. 110-111, Illinois.
24. R. GARCIA ROSA; M^a T. DE PEDRO LUCIO; C. GONZALEZ; A. ALI-QUE. "Diseño y realización de un lenguaje de descripción - de tarjetas de circuitos impresos". Actas del III Congreso Nacional de Informática y Automática, Madrid 1975.
25. J. L. SAN EMETERIO PRIETO. "Optimización de la tabla de símbolos en un lenguaje para diseño automático". Tesina. Universidad Complutense de Madrid, Facultad de Ciencias, 1977.
26. M^a T. DE PEDRO LUCIO; R. GARCIA ROSA. "DOCIL - An automatic system for printed circuit board (PCB) designing. A -- board description language and an algorithm to connect a - set of points". Proc. of the 14th Design Automation Conference. (IEEE Catalog number 77, CH - 1216 - 14). New Orleans 1977.

Referencias a las que no se alude directamente en el texto

27. K. WYATT. "PCB design-where are we and where are we going".
Electron, nº 64, pp. 51-54, Nov. 1974.
28. W. M. BOYCE. "An improved program for the full Steiner Tree
Problem". ACM Trans. on Math. Soft., nº 4, vol. 3, pp. 359-
-385, Dec. 1977.
29. M. HANAN. "On Steiner's Problem with rectilinear distance".
J. Siam Appl. Math., nº 2, vol. 14, Mar. 1966.
30. T.A.J. NICHOLSON. "Permutation procedure for minimising the
number of crossings in a network". Proc. IEE, nº 1, vol. --
115, pp. 21-25, Jan. 1968.
31. K.H. HOSKING. "A New technique for the placement of "Sol-
der-Blobs" and hence discrete components". The Marconi Re-
view. fourth quarter, pp. 169-183, 1975.
32. A.A. TIUTIN. "An improved algorithm for component disposition
on a board". Alta Frequenza nº 39, pp. 145-149, 1970.
33. R.M. KARP; A.C. MC KELLAR; C.K. WONG. "Near-optimal solutions
to a 2-dimensional placement problem". SIAM J. Comp. nº 3, -
vol. 4, pp. 271-286, 1975.

34. K.H. HOSKING. "The use of a very fast routing algorithm for printed circuit board design". The Marconi Review, pp. 207-226, third quarter, 1971.
35. C.W. GRAVES; A.B. WHINSTON. "An algorithm for the quadratic assignment problem. Management Science, n° 7, vol. 17, Mar. 1970.
36. B.W. KERNIGHAM; S. LIN. "An efficient heuristic procedure - for partitioning graphs". The Bell System Technical Journal, pp. 291-307, Feb. 1970.
37. R.L. RUSSO; P.H. ODEN; P.K. WOLFF. "A heuristic procedure for the partitioning and mapping of computer logic graphs". IEEE Trans. on Comp., n° 12, vol. C-20, Dec. 1971.
38. G.G. TIENGO. "Programma per il posizionamento di circuiti - micrologici su una piastra". Telecomunicazioni, n° 50, pp. 47-50, Mar. 1974.
39. M.A. BREUER. "A class of min-cut placement algorithms" Proc. of the 14th Design Automation Conference, New Orleans, pp. 284-290, Jun. 1977.
40. W.G. MAGNUSON. "Computer oriented circuit design". Prentice-Hall, 1969.

